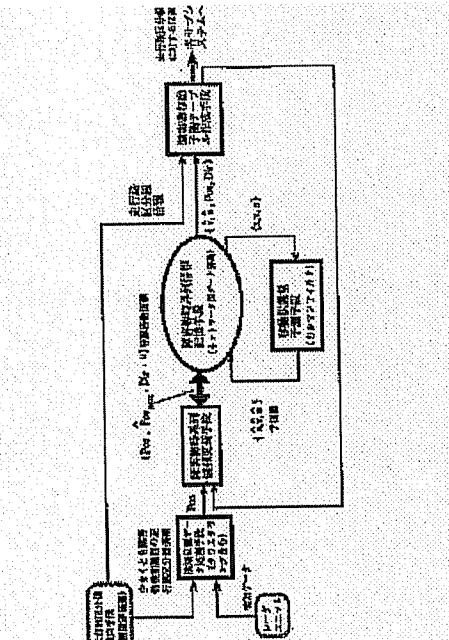


MOBILE DETECTOR**Publication number:** JP8122432**Publication date:** 1996-05-17**Inventor:** SAKAGAMI YOSHIAKI**Applicant:** HONDA MOTOR CO LTD**Classification:**

- international: B60R21/00; B60W30/00; G01S13/86; G01S13/93; G08G1/16; H04N7/18; B60R21/00; B60W30/00; G01S13/00; G08G1/16; H04N7/18; (IPC1-7): G01S13/93; B60R21/00; G01S13/86; G08G1/16; H04N7/18

- European:**Application number:** JP19940281217 19941020**Priority number(s):** JP19940281217 19941020**Report a data error here****Abstract of JP8122432**

PURPOSE: To detect the position and the moving direction of an obstacle mobile by estimating the speed of the mobile while setting an initial value equal to the average speed thereof. **CONSTITUTION:** Average speed of an obstacle is determined based on the moving amount between oldest and newest nodes in a network type data structure having nodes related to new and old information. A linear prediction (Kalman) filter estimates the predictive position of the obstacle at next detection time using the average speed thus determined as an initial value. A detected position data processing means then performs clustering for handling the road section line information and the position of the obstacle detected, respectively, through image processing and a radar unit at one node and determines the position of the obstacle with reference to an origin for each cluster. Based on a detection data being updated with time by an obstacle time series information updating means, the Kalman filter updates the origin of measurement and determines the predictive position of a next obstacle based on a moving distance (x), a speed (v) and an acceleration (a). Information related to the current position, predictive position, orientation, speed, etc., of the obstacle is then tabulated based on the predicted values.



Data supplied from the esp@cenet database - Worldwide

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-122432

(43)公開日 平成8年(1996)5月17日

(51)Int.Cl.⁶

G 0 1 S 13/93

B 6 0 R 21/00

識別記号

庁内整理番号

F I

技術表示箇所

C 8817-3D

C 8817-3D

G 0 1 S 13/86

G 0 1 S 13/ 93

Z

審査請求 未請求 請求項の数 3 FD (全 36 頁) 最終頁に続く

(21)出願番号

特願平6-281217

(22)出願日

平成6年(1994)10月20日

(71)出願人 000005326

本田技研工業株式会社

東京都港区南青山二丁目1番1号

(72)発明者 坂上 義秋

埼玉県和光市中央1丁目4番1号 株式会

社本田技術研究所内

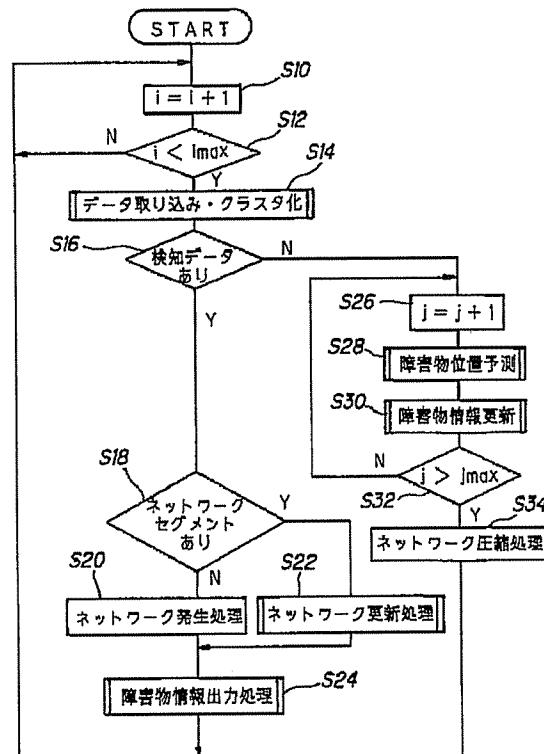
(74)代理人 弁理士 吉田 豊

(54)【発明の名称】 移動体の検出装置

(57)【要約】

【構成】 前走車など自車の走行にとって障害となる移動体の位置観測データを入力として該移動体の速度を直線運動モデルに基づいて推定するカルマンフィルタを用いると共に、その初期値として該移動体の平均速度を与える。また所定時間ごとに計算原点を更新する。

【効果】 カルマンフィルタの収束性が向上すると共に、演算量が少ない直線運動モデルを用いながら誤差の少ないリアルタイムな移動体情報を得られる。



1

2

【特許請求の範囲】

- 【請求項1】 a. 自己に対して相対的に移動する移動体の位置を検出して出力する検出手段、
 b. 前記検出手段の出力に基づき、前記移動体の所定の時点からの移動距離を算出する移動距離算出手段、
 および
 c. 前記算出された移動距離を入力として、前記移動体の速度を少なくとも推定する線形予測フィルタ、からなる移動体の検出装置であって、
 d. 前記検出手段の出力に基づき、前記移動体の所定期間における平均速度を算出する平均速度算出手段、を備え、前記線形予測フィルタは、前記算出された平均速度を初期値として前記速度を少なくとも推定することを特徴とする移動体の検出装置。

【請求項2】 前記線形予測フィルタは、前記算出された平均速度を初期値として前記移動体の初速度を推定することを特徴とする請求項1項記載の移動体の検出装置。

- 【請求項3】 a. 自己に対して相対的に移動する移動体の位置を検出して出力する検出手段、
 b. 前記検出手段の出力に基づき、前記移動体と基準位置との間の距離を算出する距離算出手段、
 c. 前記算出された距離を入力とし、直線運動モデルに基づいて前記移動体の運動パラメータを推定する手段、からなる移動体の検出装置において、所定時間ごとに前記基準位置を前記移動体の現在位置付近に更新するようにしたことを特徴とする移動体の検出装置。

【発明の詳細な説明】**【0001】**

【産業上の利用分野】 この発明は移動体の検出装置に関し、より具体的には走行時に障害となる前走車などの移動体の検出装置であって、走行環境におけるその移動体情報を精度良く検出できるようにしたものに関する。

【0002】

【従来の技術】 前走車の検出装置としては、例えば特開昭61-278775号公報記載の技術が知られている。その従来技術においては、移動する前走車について検出時刻の異なるデータ同士を関連づけて同定しているが、前走車の横方向への移動は、単に検出されるレーダーピームの変化として捉えているに過ぎないため、前走車の位置や移動方向を精度良く検出することができなかつた。

【0003】

【発明が解決しようとする課題】 従って、この発明の目的は上記した従来技術の欠点を解消することにあり、前走車など走行環境において障害となる移動体の位置や移動方向を正確に検出し、走行環境における移動体情報を精度良く検出できるようにした移動体の検出装置を提供することにある。

【0004】 更に、前走車などの移動体の位置や移動方

10

20

30

30

40

50

向を精度良く検出するには、その位置データから移動体の運動パラメータを予測する必要がある。位置データは様々なノイズを含んでおり、それから移動体の運動パラメータを予測するには、カルマンフィルタなどの予測フィルタを用いることが有効である。しかし、そのとき障害物の運動として曲線運動モデルを用いると演算量が増え、処理の高速化と装置の小型低コスト化との両立が困難となる。そこで、演算量の少ない直線運動モデルを用いると、予測フィルタへの入力として例えば移動体の位置データから求めた移動距離を使用することになるが、移動体が曲線路を走行する場合、実際には曲線運動であるため移動距離の誤差が経時に蓄積し、予測しようとする位置、速度、加速度などの運動パラメータの誤差として表れる。

【0005】 従って、この発明の第2の目的は、演算量の少ない直線運動モデルを用いつつ前走車などの移動体の運動パラメータを精度良く予測すると共に、走行環境が曲線路を含むときも、運動パラメータの誤差が蓄積しないようにした移動体の検出装置を提供することにある。

【0006】

【課題を解決するための手段】 第1の目的を達成するために、この発明は請求項1項において、自己に対して相対的に移動する移動体の位置を検出して出力する検出手段、前記検出手段の出力に基づき、前記移動体の所定期間における平均速度を算出する平均速度算出手段、を備え、前記線形予測フィルタは、前記算出された平均速度を初期値として前記速度を少なくとも推定する如く構成した。

【0007】 第1の目的を達成するために、この発明は請求項2項において、前記線形予測フィルタは、前記算出された平均速度を初期値として前記移動体の初速度を推定する如く構成した。

【0008】 第2の目的を達成するために、この発明は請求項3項において、自己に対して相対的に移動する移動体の位置を検出して出力する検出手段、前記検出手段の出力に基づき、前記移動体と基準位置との間の距離を算出する距離算出手段、前記算出された距離を入力とし、直線運動モデルに基づいて前記移動体の運動パラメータを推定する手段、からなる移動体の検出装置において、所定時間ごとに前記基準位置を前記移動体の現在位置付近に更新する如く構成した。

【0009】

【作用】 請求項1項においては、検出手段の出力に基づき、前記移動体の所定期間における平均速度を算出する平均速度算出手段、を備え、線形予測フィルタは、前記

算出された平均速度を初期値として速度を推定する如く構成したので、自車の走行にとって障害となる前走車などの障害物の位置や移動方向を精度良く検出することができる。それによって、警告、回避など必要な動作を的確に行うことができる。尚、ここで「線形予測フィルタ」は周囲の状況に適応して動作するフィルタを意味し、実施例ではカルマンフィルタを用いた。また、平均速度としたが、そうでなくとも良いことは言うまでもない。

【0010】請求項2項においては、前記線形予測フィルタは、前記算出された平均速度を初期値として前記移動体の初速度を推定する如く構成したので、線形予測フィルタの収束性が向上して推定精度を一層上げることができる。

【0011】請求項3項においては、所定時間ごとに基準位置を前記移動体の現在位置付近に更新する如く構成したので、演算量の少ない直線運動モデルを用いながら、走行環境が曲線路を含むときも、自車の走行にとって障害となる前走車などの移動体の運動パラメータを誤差なく予測することができる。

【0012】

【実施例】以下、添付図面に即してこの発明の実施例を説明する。

【0013】図1はこの発明に係る移動体の検出装置を備えた車両を全体的に示す説明透視図である。図において、車両はCCDカメラ（モノクロTVカメラ）10を1基備える。CCDカメラ10は運転席上方のルームミラー取り付け位置に固定され、車両進行方向を単眼視する。符号12はミリ波レーダからなるレーダユニットを示し、車両前方に取り付けられた2基の前方レーダからなり、反射波を通じて他車などの移動体を含む立体障害物の存在を検出する。

【0014】レーダユニット12は、より具体的にはF-M-CW型のレーダ装置からなり、図2に示す如く、複数本の電磁波ビームを送波し、その反射波を受信して立体障害物の存在を検出する。検出された障害物の位置座標（相対距離）は、ビームごとに発生する。尚、このレーダユニットの詳細は、本出願人が先に提案した特願平5-276396号に詳細に述べられているので、これ以上の説明は省略する。

【0015】図1に戻ると、車両室内の中央部付近にはヨーレートセンサ14が設けられ、車両の重心を通る鉛直軸（z軸）回りの車両回転角（ヨ一角）の変化速度（ヨ一角速度）を検出する。更に、車両のドライブシャフト（図示せず）の付近にはリードスイッチからなる車速センサ16が設けられ、車両の進行速度を検出すると共に、舵角センサ18が車両のステアリングシャフト20の付近に設けられてステアリング舵角を検出する。ステアリングシャフト20には舵角制御モータ22が取り付けられると共に、スロットル弁（図示せず）にはバル

スモータからなるスロットルアクチュエータ24が取り付けられ、更にブレーキ（図示せず）にはブレーキ圧力アクチュエータ26（図1で図示省略）が取り付けられる。この構成において、車両は算出された舵角制御量に応じて舵角制御されると共に、スロットル弁が閉鎖されて車速が調節され、また必要に応じてブレーキが作動させられて走行する。

【0016】図3は上記の構成をより詳細に示すプロック図で、図4は図3の構成をより機能的に示す説明図である。

【0017】図3を中心に説明すると、CCDカメラ10の出力は、画像処理ハードウェア30に送られてエッジ検出とハフ（Hough）変換により直線成分の抽出が行われ、その結果はバス32を介して通信用メモリ34にストアされる。画像処理CPU36は、直線成分から走行路（道路）区分線候補を抽出して通信用メモリ34にストアする。画像評価CPU38は、所定時刻ごとにストア値を読み出して走行路（道路）区分線を決定する。レーダユニット12の出力は、レーダ処理回路40およびバス32を介して通信用メモリ34にストアされる。レーダ評価CPU42は、所定時刻ごとにストア値を読み出して障害物の位置を座標上で検出する。

【0018】また車速センサ16などの出力は、軌跡推定CPU44に送られて自車両の移動軌跡が推定される。行動計画意思決定CPU50は、前記ストア値から目標経路を作成する。その目標経路と推定された自車の移動軌跡は、軌跡追従制御CPU46に送られ、そこで軌跡（目標経路）追従制御量が決定される。更に、軌跡追従制御CPU46は、舵角制御量を算出して舵角制御CPU52に出力する。舵角制御CPU52は、PWMコントローラ54およびドライバ56を介して前記舵角制御モータ22を駆動する。尚、モータ駆動量はエンコーダ58を通じて検出され、フィードバック制御が行われる。

【0019】また行動計画意思決定CPU50は、その速度・追従制御部で車体の目標加速度を求め、車速制御CPU60に送出する。車速制御CPU60は、アクセルパルスモータコントローラ62、ドライバ64を介してスロットルアクチュエータ24を駆動すると共に、ブレーキソレノイドコントローラ66およびドライバ68を介してブレーキ圧力アクチュエータ26を駆動する。その駆動量は圧力センサ70を介して検出され、第2のフィードバック制御が行われる。

【0020】上記において、レーダ評価CPU42が主として、実施例に係る移動体の検出装置に相当する。

【0021】図5は実施例に係る移動体の検出装置の動作を示すメイン・フロー・チャートである。また、図6はその動作を説明する機能プロック図である。

【0022】図5フロー・チャートを参照する前に、実施例に係る移動体の検出装置の動作を概説する。

【0023】実施例に係る装置において、FM-CW型のレーダユニット12により時々刻々と検知された、前走車などの移動体を含む障害物の位置座標に自車の現在位置（座標軸とヨー角）を付与し、障害物の絶対座標を保持しながら、過去の系列から、次の検知時刻における位置をカルマンフィルタ（線形予測フィルタ）により予測し、同一のものかどうか判断する。従来の車間距離を計るレーダでは、ある時刻に検知されて次の時刻に検知されたものは同一であると言う前提を置いているが、その障害物が車両であるのか、どちらの方向に向かって進んでいるのか分からぬため、警報などで誤ることがある。実施例に係る手法では、移動する障害物のおよその方向、速度、加速度を検出し、よって危険の予知と共に、的確な警報、自車の制御などを行うことを可能とした。

【0024】移動する車両などは通常の走行状態では、その軌跡は滑らかであり、速度も過度に急変することはないので、その性質を用いて検知した障害物の特性を表現することができる。実施例では後述の如く、それを「移動体らしさ」と表現する。それによって、障害物の位置しか検出できないレーダユニット12を用いても、その動きを一意に表せ、自車の行動制御に利用できる。例えば、手前で一時に検知された障害物が更に接近したところで再度検知されたとき、後述の如く、同一のものと見做すことができ、それが停止している可能性が高い、前述した表現で言えば移動体らしさが低いならば、その側方を通過できる余地があるときは減速して通過する、余地がないときはその前で停止するなどの制御を行うことができる。

【0025】この実施例においては後述の如く、時系列に発生した複数障害物の情報を効率良く管理し、新たに入ってきた情報とそれまで保持していた情報を比較し、時間空間的に関連するものをノードとそれらを結ぶリンクとして表現する、いわゆるネットワーク型のデータ構造を用いる。それにより、複数障害物の情報の内容を更新する手続きが容易になる。実施例では現在検知されている1つの障害物に関するデータの一連を「セグメント」と呼ぶ。後述の如く、移動体らしさを評価しつつセグメントを更新することにより、複数の障害物に対しても移動情報を簡単に求めることができる。前記のネットワークセグメントの情報から、次に移動する位置が推定でき、入力データが一時的になくなってしまっても、関連性の高いデータが再度検知されれば、そのデータはノードとしてセグメントにリンクされて再捕捉することができ、移動情報が途中で失われることがない。

【0026】以上を前提として、図5フロー・チャートを参照して実施例に係る装置の動作を説明する。尚、図5フロー・チャートに示すプログラムは、レーダ検知サイクルの100msごとに起動される。

【0027】先ずS10でセグメントの個数をカウント

するカウンタ1の値を1にインクリメントし、S12に進んで最後のセグメントimaxに達していないことを確認してS14に進み、そこでデータ取込み・クラスタ化を行う。

【0028】実施例で使用するFM-CW型レーダの特性上、入力データから障害物の幅を検知することができず、送波ビームの照射広がり角、障害物までの距離によつては同一障害物に対して複数の障害物があるかのように検知される。そこで、ある基準でまとめてクラスタとして扱う。その基準は、クラスタの幅と縦の長さをしきい値として、図7に示すように、普通乗用車のサイズを用いる。クラスタの幅を障害物の幅とし、重心を中心とし、クラスタを後述するネットワークセグメントのノードとして扱う。その操作により、後の処理でのデータ量を抑えることができる。更に、どのビームに検知されているかを検知の度に意識する必要がないと言う利点もある。

【0029】車両や道路外の物体など複数の障害物が存在する場合、それらが道路に対してどの位置にあるのかが分かれば、適切な警報動作や自車の制御が可能となる。しかしながら、図8に示すようにカーブ路（曲線路）を通過するときなど、前走車など自車の進路上に存在する障害物と樹木など道路外に存在する障害物とが、明確に識別し難い。この場合、従来技術は例えば特開平4-193641号公報に示されるように、撮像信号の中から自車線または道路形状を示す信号を抽出して自車の進路を判別すると共に、撮像された範囲に存在する物体の位置を測定し、測定された物体の中で自車の進路上に存在する物体を障害物と判断している。

【0030】このように、特開平4-193641号公報記載の技術は、進路の中か外かで障害物になるか否か判別している。これにより、樹木などの静止障害物は障害物ではないと判別することができるが、車両などは時々刻々と移動することから、この従来技術ではそれが自車にとって障害物になるか否かを必ずしも的確に判別することができない。

【0031】即ち、レーダデータをそのまま使用しているので、必ずしも実際の障害物と一对一には対応しない。何故ならば、1つの障害物の異なる部位から別々にレーダ波の反射があり、複数の検出データが出力されることがしばしばあるからである。このため、上記した従来技術によるときは、実際の障害物の正確な位置を知ることができない。また、検出データは点の情報であるので、実際の障害物の大きさを掴むことができない。

【0032】そこで、この出願に係る実施例においては、画像処理によって検出された走行路（道路）区分線の情報とレーダによって検知された障害物の位置とを比較し、ある基準を用いて障害物としてデータを1つに扱う（クラスタ化する）ようにした。その場合に問題となるのが、走行路の外の物体（例えば、道路標識、ガード

レール）に起因する検出データである。検出したい障害物（他の車両など）が、このような走行路外の物体の傍らに位置していると、障害物に起因するデータと一緒に走行路外の物体に起因する検出データまでもが同じクラスタに併合される。

【0033】実施例ではクラスタは検出データを距離の近いもの同士をまとめて大きさと中心位置を持つものと認識することとしたが、走行路外の物体まで同じクラスタに併合されると、障害物の大きさや中心位置を正しく求めることができない。そこで、図9に示す如く、画像処理で得た走行路区分線で分断されるか否か判断し、分断されるならば複数のデータを別々のものとして扱い、道路内と外に障害物があると判断した。それにより、図9左側に示すような1つの障害物と判断する不都合が生じない。

【0034】図10はその作業を示すサブルーチン・フロー・チャートである。以下説明すると、S100でカウンタm（クラスタ内の検知データ数カウンタ）、n（検知データ数カウンタ）を0に、j（クラスタ数カウンタ）を1にリセットする。またフラグf1gを0にリセットする。

【0035】続いてS102に進み、検知データ数カウンタnをインクリメントし、S104に進んで当該nの検知データDnをその時点で注目している検知データDとする。即ち、いま図11の状態にあるとし、検知データD1, D2, D3, D4のうち、D1に注目する。続いてS106に進んで前記フラグf1gが0であるか否か判断する。このフラグは新たにクラスタを新設するとき0にリセットされるものであり、先にS100で0にリセットされていることから、ここで判断は肯定され、S108に進んでDが最終検知データか否か判断する。

【0036】図11の状態にあると仮定すると、S108での判断は否定され、S110に進んで変数L, P, Gを検出データDに書き替える（図12）。ここで、Lはクラスタ内で最もy座標値の大きい検知データ、Pはその時点で最後にクラスタに併合した検知データ、Gはクラスタ内の検知データの座標値の総和を意味するが、ここではD1をとりあえず、y座標値が最も大きく、最後にクラスタに併合したものであり、それを総和としておく。同時に、フラグf1gの値を1にセットし、クラスタ内検知データ数カウンタmを1に書き替える（即ち、最初のクラスタ内に検知データが1つあるとする）。

【0037】続いてS102に戻り、検知データ数カウンタnをインクリメントし、S104に進んでその時点で注目している検知データをD2とし、S106に進んでフラグの値が0か否か判断する。S110を経てきたことから今度は否定されてS112に進み、図13に示すように、先の検知データ（D1）と現在の検知データ（D2）のx, y方向の差（絶対値）Lx-Dx, Ly-Dyが所定値（ここでは「Wth」（例えば5m）と「Dth」（例えば2m）とする）より共に小さいか否か判断する。所定値Wth, Dthは、実際の乗用車の大きさ、即ち、z軸から見た投影面積には相当する。

【0038】S112で否定されたとするとS114に進み、線分PD、即ち、D1(P)とD2(D)の間に引かれた線分と走行路区分線（画像処理を経て実線で示される）が交わるか否か判断する（図14）。先に述べた如く、カーブ路を走行するときなど、道路内と外の物体が同時に検知され、それらが自車の走行に際して障害物になるか否か、そのままでは判断することはできない。そこで実施例においては、検知データが複数個あるとき、クラスタ化すると共に、検知データ間に走行路区分線が存在するときは、そこで別のクラスタとする。かく構成することによって、自車の走行にとって真に障害物となる物体を正確に判別することができる。

【0039】図14に示す例では交わらないことから、ここで判断は否定され、S116に進んで検知データD2をPに書き替えてその時点で最後にクラスタ1（S100で特定）に併合した検知データとし、D(D2)の座標値を加算して総和Gを更新し、D(D2)をクラスタ1内で最もy座標値の小さい検知データとする。同時にクラスタ1内の検知データ数カウンタ値mをインクリメントし、クラスタ1内に検知データが2個あるとする。

【0040】続いてS118に進んでD(D2)が最終検知データか否か判断する。ここで、最終か否かはy軸に関して判断されるが、S118の判断は図14の例では否定され、S102に戻り、S104に進んで今度は検知データD3に注目し、S112を経てS114に進み、新たな検知データD3と最前の検知データD2の間に引かれた線分PDと走行路区分線が交わるか否か判断する。図15に示すように、この場合は肯定されてS120に進む。

【0041】尚、S112で否定されるときは、S114で線分と走行路区分線が交わるか否かを判断するまでもなく、S120に進む。S120以降はクラスタを別にするための処理であるが、検知データ間の距離が車両の大きさを超えるときは、別のクラスタとする。即ち、実施例で使用するレーダの特性から同一の車両に対して複数の検知データが生じる可能性があるため、S114を介して確認するものであるが、検知データ間の距離が車両の大きさを超えるときは、別の物体と判断するのが妥当と考えられるからである。

【0042】S120ではクラスタ内検知データ数カウンタ値mが1か否か判断する。即ち、新たなクラスタの検知データが1個であるか否か判断する。図示例の場合には先にS116で2にインクリメントされていること

から否定され、S 1 2 2 に進んでクラスタ 1 内の検知データの座標値の総和 G をクラスタ内検知データ数カウンタ値 m (2) で除算してクラスタ 1 の重心位置 G_j (j = 1) を求める。同時に、L (D 1), R (D 2) 間の距離の絶対値を求めてクラスタ 1 の幅 W_j とする。

【0043】また、ここまでで検知データ D 1, D 2 をクラスタ 1 とし、D 3 から別のクラスタとするためフラグの値を 0 にリセットし、クラスタ内検知データ数カウンタ値 m をインクリメントする。続いて S 1 2 4 に進んでクラスタ数カウンタ値 j をインクリメントする。尚、S 1 2 0 でクラスタ内の検知データが 1 と判断されるときは S 1 2 6 に進んでその検知データの位置 L を重心位置 G_j とともに、検知データが 1 個でクラスタとしての幅がないことから値 W_j は求めず、S 1 2 8 に進んでフラグの値を 0 にリセットする。

【0044】続いて S 1 0 8 に進み、最終検知データか否か判断する。図示では D 3 であったことから否定され、S 1 1 0 に進んで図示の処理を行い、S 1 0 2 以降を同様に進み、S 1 1 8 で D が最終検知データと判断されるまで以上のステップを繰り返す。そして、S 1 1 8 で最終検知データと判断されるときは S 1 3 0 に進んで検知データ数 m が 1 か否か判断し、否定されるときは S 1 3 2 に、肯定されるときは S 1 3 4 に進んで先に S 1 2 2, 1 2 6 で述べたと同様の処理を行って終わる。また、S 1 0 8 で肯定されるときは検知データが 1 個であったことから、S 1 3 6 に進んでその検知データを重心位置に書き替えて終わる。

【0045】図 5 フロー・チャートに戻ると、続いて S 1 6 に進んで検知データがあるか否か判断する。肯定されるときは S 1 8 に進んでネットワークが既に存在するか否か判断し、否定されるときは S 2 0 に進んでネットワーク発生処理を行う。ここで、先に触れたネットワークについて詳説する。

【0046】レーダによって時々刻々検知されるデータは、それをつなげていけば、位置に関する時系列データとなる。実際の道路（走行路）上では、他の走行車両や停止している車両、ガードレールや標識など複数の物体が存在するために、同時に別のものが検知される。よって、これを切りわけ、各々のおよその移動速度、移動方向を求める必要がある。そこで、実施例の場合、同時に複数の障害物が扱えて、かつレーダユニットの検知誤差を考慮したデータ構造とし、それらの更新を新しい入力で過去までの累積情報にどれだけ依存しているか、即ち、関連性を持つかを移動予測による位置推定に基づいて行うこととした。

【0047】S 2 0 の作業は、このネットワーク型のデータ構造を新設する作業を意味する。具体的には、図 1 6 に示すように、時刻 i における障害物情報をノードとする、移動体全体に関する時系列情報を格納したネットワークセグメントを作成する作業である。ノードには図

1 7 に示す情報を持たせる。セグメントには図 1 8 に示す情報を持たせる。以上のようなデータ構造によって、検知されたデータを内部に保持する。移動する物体が車両の場合、一般的な状況では車線変更を開始してから終了するまでに約 2 秒を要する。よって、車両の動きをとらえるのに十分なように約 2 秒分のデータが保持できるノードの数を設定する。また、およその速度と方向は、ネットワークを利用して、以下の計算により簡単に求められる。

速度 = (最新ノード位置 - 最古ノード位置) / (最新データ検知時刻 - 最古データ検知時刻)

方向 = 最新ノード位置と最古ノード位置を通る直線の傾き

【0048】通常のレーダ、例えばドップラーレーダでは、対象物の相対速度が検知可能であるが、実施例で用いている、位置のみを検知する FM-CW 型のレーダでは、時間微分により速度を求めることになる。しかし、外乱の多い環境では、単純に位置座標を微分することはできない。そこで、実施例の場合、最古と最新のノード間の移動量と時刻からおおよその速度を求めた。これにより、ばらつきの影響はセグメントの始点と終点にのみ現れ、比較的安定した結果が得られる。また、方位についてもセグメント全体の向きを扱うので、ばらつきの影響は少ない。尚、ここで求めたおおよその速度は後述の如く、カルマンフィルタによる予測の初期値として用いる。

【0049】一般的な走行では 0.2 G 程度の減速を考慮すれば良く、0.2 G で 2 秒間減速した場合、実施例では推定速度と実際の速度の差は、理論上 7 km/h となって、カルマンフィルタによる予測の初期値として適当である。図 1 9 に推定速度、推定方位を示す。追従中の前走車の速度は自車の速度と比較して比較的安定している。但し、停止障害物について、レーダのばらつきの影響を受けている。これは、速度の計算が先のネットワークセグメントの始終点間距離に基づいているためである。

【0050】以上のネットワークセグメントを検知している障害物の数だけ作成し、コンピュータ内部に保持し、新たに検知されたデータに基づいてネットワークを更新する。

【0051】図 5 フロー・チャートの説明に戻ると、S 1 8 でネットワークが既に存在すると判断されるときは S 2 2 に進み、ネットワーク更新処理を行う。

【0052】これについて述べると、ネットワークの更新を単純更新と複数障害物存在時の更新に分類し、分類結果により処理を違えるようにした。

1) 単純更新

新たに検知された障害物のデータがどのセグメントに属するかを推定位置との距離に基づいて判断する。即ち、その距離がレーダシステムの検知ばらつき (x 变位と y 变

位)より求めた範囲にあれば、セグメントに属し、なければ、新たな別の障害物として扱う。即ち、新しいセグメントを発生させる。更新の条件、処理方法については図20に示す。

2) 複数障害物存在時の更新

一般的の道路において例えば、前走車、追い抜き車両、路肩に停止した車両、標識など様々な障害物が存在し、レーダにも検知される。実際の場面では、前走車の側方に他の車両が走行していたり、同一車線の前方に複数の車両が走行していて、そのうちの1台が隣の車線に移ったり、追い越した車両が追従している前走車の前あるいは後ろへ進入するなど種々の場合を考えられる。このような複雑な状況におけるネットワークの更新は、以下の3形態で行う。

a) 新規発生

新たに別の障害物が検知されたとき、新しいセグメントを発生させる。

b) 分岐継承

同じような動きをしていた複数の障害物が別の方へ向きを変えたとき、過去の経緯の情報（ノード情報、セグメント情報）を新しいセグメントに継承させる。

c) 合流消滅

障害物同士が近づき、どちらか一方しか検知されなくなったとき、移動体らしさと言う評価基準に基づいて一方を削除する。

それぞれの更新条件、処理方法を図21に示す。

【0053】そして、複数障害物が存在するときの合流消滅時の更新において、一方を削除するために前記した移動体らしさと言う評価基準を設ける。これは、移動する過程、検知データの特性を考慮した評価基準である。その要素として、移動推移、検知時間、センサ検知範囲、障害物位置を図22のように考え、移動体らしさは図23に示すように算出する。この表現によれば、停止物体（障害物）についても移動体と全く同じに扱うことができる。

【0054】図24に自車が走行中に検知した障害物の移動体らしさの推移を示した。時刻40（×50ms）において検知されたデータは、前走車（実線）と停止障害物（破線）が同一に扱われたが、移動している前走車の移動体らしさが相対的に大きいため、停止障害物のセグメントは消滅した。

【0055】以上の手続きにより、ネットワークセグメントの情報を更新する。即ち、図18の内容を計算する。ノードが所定の数（約2秒分）を超えたたら、各値から最古のノードに関する移動量や曲率、これらの自乗和などを減じ、新たなノードの情報を加える。これにより、セグメントは、新しい入力の度に更新され、セグメントノードの一括計算を要しないので、複数の障害物があっても十分に短い処理時間で扱うことができる。

【0056】図25はそのネットワークの更新作業を示

すサブルーチン・フロー・チャートである。以下説明すると、S200でノード接続テーブルC_{tx}[j]を初期化する。即ち、ノード接続テーブルC_{tx}[j]=0とする（j=1～検知データクラスタ数）。次いでS202に進んで接続判定用テーブルの配列を初期化、即ち、M_{tx}[i][j]=0とする（i=1～最大セグメント数）。

【0057】次いでS204に進んでセグメント数カウンタ値iをインクリメントし、S206に進んでセグメント[i]のノード数NNがしきい値N_{th}（例えば5（500ms））を超えるか否か判断する。否定されるとき、即ち、データが500ms分溜まっていると判断されるときはS208に進んでフラグf_{1gn}の値を0にリセットすると共に、500ms分溜まっていると判断されるときはS210に進んでフラグf_{1gn}の値を1にセットし、S212に進んで障害物位置予測を行う。尚、ここで500ms分溜まっていないと位置予測を行わないのは、時系列データがこの程度蓄積されないと、予測データの信頼性が低く、当初の目的にそぐわないからである。また500msは、レーダの検知サイクルで言えば、5回分に相当する。

【0058】図26はその作業を示すサブルーチン・フロー・チャートであるが、同図を参照して説明する前に、その背景について説明する。

【0059】高速道路上の車両を想定すると、車両は急旋回せず、車線変更は二状態予測制御に基づくと言う前提条件に立てば、移動はほぼ直線的として扱うことができる。よってセグメントのノード位置を回帰直線で近似することが可能となる。即ち、車両は2秒の間この直線上を進んだと仮定する。更に、移動量は後述のようにカルマンフィルタにより予測し、次の検知時刻にその車両が移動している位置を先の直線上に求める。

【0060】実際は、セグメントの最新ノード通り、セグメントの近似直線と直交する点から現在時刻（新たに検知された時刻）までに進む距離だけセグメントの近似直線上を進めた点Pを求める（図27）。この位置を推定位置、この作業を移動予測と呼ぶ。この手法によると、現実に、運動モデルに直線移動、速度可変のものを適用したことと同等である。

【0061】実際に移動する車両と停止障害物における推定位置（実線）と入力位置（鎖線）をプロットしたものを図28、29に示す。移動する車両についてはほぼ一致し、停止障害物では約2mのずれ程度であり、移動障害物と停止障害物が近くにあると、区別できない場合が生じる。これは、後述の複数障害物存在時の処理で対処する。また、レーダは、障害物の形状や向きによっては、一時的に検知不能時間が発生する。即ち、短時間見失うことがある。この移動予測によれば、新たな時刻に基づく予測位置を出すので、短い時間ならば、障害物の再捕捉が可能である。

【0062】ところで、自動運転車の制御系では、画像処理、本処理、行動計画などがそれぞれ非同期に処理を行う場合、データの受渡しタイミングが異なる。即ち、データを受ける側の処理時間に依存してデータの遅れ時間が発生する。実施例では、障害物の検知時刻における速度、加速度、方位から運動モデル（物理モデル）に従い、カルマンフィルタ（前記した線形予測フィルタ）を用いて任意時間先の位置を計算し、データを受ける側の処理の時刻を基準とした障害物の情報が得られるようにした。これにより、処理時間の異なるシステム間で通信遅れ時間を意識しないで障害物の情報を扱うことができる。

【0063】但し、カルマンフィルタの処理結果により推定された速度、加速度はゆらぎがあり、その影響を受けるため、任意時間の設定は余り大きくできない。実施例では、レーダ検知時刻から1秒間を100msごとに障害物の位置を算出している。図30は前走車が定速から減速して停止する際の実際の前走車の速度（破線で示す）、本処理における推定速度、1秒間の予測位置における速度をプロットしたものである。減速時には、途中から推定速度が実測値を下回っているが、処理時間、通信遅れ時間、および制御時間の遅れを考慮して、この予測値を使うことにより、システム全体の遅れを減少させることができる。尚、予測時間と計算時間刻みを任意に変更しても良い。

【0064】処理結果は、図3に示す行動計画意思決定CPU50、画像処理CPU36、車速制御CPU60の各サブシステムに通信用メモリ34経由で渡される。各サブシステムは、処理時間が異なるため、基本的に非同期である。このことは、レーダが障害物の位置を検知した時刻から、実施例の処理を行った後、実際に各サブシステムへデータが渡るまでに時間の遅れがあることを意味する。実施例では、障害物の現在の速度、加速度、移動方位に基づいて、任意の先の時刻を求めるが、その予測情報（後に「障害物移動予測テーブル」として示す）を渡すことにした。その結果、各サブシステムは、軌跡推定からデータを受けた時刻に基づいて、障害物移動予測テーブルの参照位置を決定し、データを使用することができる。これにより、通信時間の遅れを意識しないで、障害物の情報を扱うことができる。

【0065】その予測をカルマンフィルタを用いて行うので、次にカルマンフィルタについて説明する。

【0066】カルマンフィルタは、時系列ある信号 $x(t)$ が白色信号によってドライブされた線形ダイナミカルシステム（確率差分方程式）の出力として、信号の相関性がダイナミックスの構造に由来すると見做し、線*

$$P(k+1 | k+1) = P(k+1 | k) - W(k+1) S(k+1) W'(k+1)$$

【0078】

【数7】

*形予測フィルタを構成する。これにより、信号や雑音の時系列を実時間で扱うことができると言う特徴がある。

【0067】宇宙開発における衛星の軌跡推定では、ペイズの重みつき最小2乗フィルタ（バッチフィルタ）が古くから使用されており、現在も使用されている。バッチフィルタは、データを一括処理することから、ある時点からある時点までの予測を一度で行う。一方、車載する障害物検知システムにおける障害物の移動推定は、検知してからの予測、認識、判断に要する時間が、短かなければならぬ。通常のバッチフィルタでは遅れがあり、衛星のような遠距離にある移動物体の推定には十分使えるが、衝突の回避などを行うには、その遅れは致命的なものとなる。また、カルマンフィルタは、常に最新のデータを用いてリアルタイムに計算でき、専用ハードウェア化しやすいと言う利点がある。

【0068】実施例では、カルマンフィルタの構成は以下の通りとした。

【0069】離散時間系のプロセスは、数1のように表すことができる。

20 【0070】

【数1】

$$x(k+1) = F(k)x(k) + G(k)u(k) + v(k)$$

【0071】ここで、 $x(k)$ は時間 k における状態ベクトル、 $u(k)$ は入力、 $G(k)$ は既知の行列、 $v(k)$ は平均値0、共分散行列 $Q(k)$ のガウシアンノイズである。

【0072】また、計測過程は、数2のように表すことができ、それに基づいてカルマンフィルタの計算ステップは数3から数8のように示すことができる。ここで、30 $w(k)$ は平均値0、共分散行列 $R(k)$ のガウシアンノイズである。初期値 $x(0)$ は、平均値 x 、共分散行列 P のガウシアンノイズとする。

【0073】

【数2】

$$z(k) = H(k)x(k) + w(k)$$

【0074】

【数3】

$$P(k+1 | k) = F(k)P(k | k)F'(k) + Q(k)$$

【0075】

【数4】

$$S(k+1) = H(k+1)P(k+1 | k)H'(k+1) + R(k+1)$$

【0076】

【数5】

$$W(k+1) = P(k+1 | k)H'(k+1)S(k+1)$$

【0077】

【数6】

$$\hat{x}(k+1 | k) = F(k)\hat{x}(k | k) + G(k)u(k)$$

【数8】

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1)(z(k+1) - H(k+1)\hat{x}(k+1|k))$$

【0080】実施例では、図31に示すように、状態ベクトル $x(k)$ の要素をそれぞれ、距離（計測原点（起点）からの移動距離）、速度、加速度とし、系を等加速度直線運動とする。従って、計測は障害物発見時からの距離とする。

【0081】ところで、カルマンフィルタの計算には初期値を与えるなければならない。その初期値は、実際の値に近いほど、より真値に近づく。図32は、真値 50 km/h に対して初期値に自車の速度 100 km/h を与えた場合である。収束までおよそ 1.7 秒を要している。実施例では先に述べた如く、およその速度を時系列データのネットワークから求め、カルマンフィルタの初期値として与えるようにした。その結果、図33に示す如く、収束性を向上させることができた。

【0082】レーダが障害物を検知した時刻から、初めて速度推定値を出力するまでの時間を、どれくらいに設定すべきかと言う問題がある。これは、実際のテストから約 0.6 秒（レーダのサンプリング周期で6回）とした。レーダの揺らぎによってこの時間は変動する。即ち、レーダが十分に精度が良ければ少ない時間で良く、逆に精度が悪い場合は、時間を長くとる必要がある。また、ネットワークによる推定誤誤を考慮すると、 0.6 秒間のネットワークによる速度の誤差は障害物の加速度に応じて 1.8 km/h (0.1 G) から 9.0 km/h (0.5 G) であり、カルマンフィルタへの初期値としては問題ない。

【0083】実施例では、カルマンフィルタの系のモデルに、等加速度直線運動の物理モデルを用いている。障害物を最初に発見してから（計測原点から）の移動距離に基づいて、レーダユニットの出力時間ごとにカルマンフィルタの計算を行っている。ところが、実際の車は、カーブなどを通過するため、直線運動はしない（図34）。これは、車が長時間にわたってカーブを通過したり、トラックコースのようなスタート地点に戻るような場合（一般道路ではそのようなことはほとんどないが）、カルマンフィルタの計算が時間が経つにつれて、誤差が増えることになる。その誤差を増やさないために、計算スパンを設定し、ある時間ある個数のデータを処理したら、計測原点を最新検知位置付近に更新するようにした。尚、計算スパンは、誤差分散が収束する時間以上を設定すれば良いことが実験の結果判明した。その結果から、実施例では2秒とした。

【0084】図35に原点更新しない場合とした場合の、カーブ通過中の前走車（ 90 km/h の定速走行）の推定結果を示す。原点更新しない場合は時間の経過と共に、推定速度が低下している。原点更新した場合は、推定速度の低下が抑えられている。原点を更新せずにこ

のような問題に対応するには、非線形なモデルを用いたカルマンフィルタを用いれば良い。しかし、非線形なモデルを使用すると複数の障害物についての処理時間が長くなり、リアルタイムに予測するには適当ではない。原点を更新すれば、線形モデルを使用しても誤差の蓄積を防ぐことができる。

【0085】以上を前提として、障害物位置予測について、図26フロー・チャートを参照して説明すると、S300で上記したカルマンフィルタを用いて図示の如く予測値を計算する。ここで、 \hat{x} 、 \hat{y} が、図6ブロック図の $P_{os} \hat{x}_{next}$ に相当するが、それについては最後に述べる。

【0086】図25フロー・チャートに戻ると、続いてS214に進み、クラスタごとの接続ノード数テーブル値 $S_{tx}[i]$ を0にリセットし、S216に進んで検知クラスタカウンタ値 j をインクリメントし、S218に進んでフラグ f_{1gn} の値が0であるか否か判断する。肯定されるとき、即ち、データが 500 ms 分溜まっているないと判断されるときはS220に進み、最短距離に基づいてノードリンクの判定処理を行う。

【0087】これは具体的には、セグメント終端と障害物との距離を求めて判定結果mを得る作業である。図36の上部にその作業を示すが、ノードリンクの判定については図示の如く約束する。即ち、セグメント終端ノードと検知データとが同一位置にあるときを「停止」、 50 cm 以内のときを「単独」、 50 cm を超えてもレーダ検出誤差 X_{er} 内にあるときを「接続」、それ以外を「非接続」とする。

【0088】またS218で否定されるときは、ノード数が 500 ms を超えていることからS222に進み、障害物予測位置に基づいてセグメント終端との距離を同様に求めて行う。これは、予測位置と検知データの距離によって判定結果mを得る作業である。図36の下部にその作業を示す。

【0089】ここで、距離 $d=0$ ならば「停止」、距離 $d \leq 50\text{ cm}$ ならば「単独」とする。また距離 d を予測位置との距離 e_d と比較するとき、

$d > e_d$ の場合、 $50\text{ cm} < e_d <$ レーダ検出誤差 X_{er} ならば「接続」

$e_d \geq X_{er}$ ならば「非接続」

$d \leq e_d$ の場合、 $50\text{ cm} < d < X_{er}$ ならば「接続」

$d \geq X_{er}$ ならば「非接続」

とする。

【0090】次いでS224に進んで求めた判定結果mが“非接続”か否か判断する。これは図36に示すように、セグメント終端と検知データとが接続されるか否かを判断することである。

【0091】否定されるときはS226に進んで各テーブル値のクラスタ数C_{tx[j]}とセグメント数S_{tx[i]}とをインクリメントし、S228でクラスタjが最大クラスタj_{max}に達したと判断され、S230でセグメントiが最大セグメントi_{max}に達したと判断されるまで、上記の処理を繰り返す。

【0092】次いで図37のS232に進んでクラスタ数カウンタ値jをインクリメントし、S234に進んでクラスタテーブル値C_{tx[j]}が1を超えるか、即ち、クラスタjが複数のセグメントに接近するか否か判断する。肯定されるときはS236に進んでセグメント数カウンタ値iをインクリメントしてS238に進み、そこで接続判定用テーブル値M_{tx[i][j]}が接続、単独または停止か否か判断し、肯定、即ち、非接続状態にあればS240、S242に進んでクラスター・セグメント間の距離が最小となる組み合わせを探す。

【0093】即ち、あるクラスタjが複数のセグメントと接続可能な場合は、最短距離のものと接続すると見做す。尚、S234で否定されるときはS244に進み、そこでノード接続テーブル値C_{tx[j]}の値が0か否か判断し、肯定されるときはS246に進んでネットワークを新たに作成する。上記をS248で最大クラスタj_{max}に達したと判断されるまで繰り返す。次いでS250からS260の処理を行い、セグメントに接続しないときはセグメント終端から最も近いクラスタを選択し、そのセグメントに接続させる。

【0094】次いで図38のS262からS268を進み、それまでに選択されなかったクラスタについて新たにネットワークを発生させる。即ち、図39に示すような処理を行う。これは、先に図21で示した「分岐継承」に相当する処理であり、セグメントの過去情報をコピーした上で新たなセグメントとして登録する。次いでS270からS280を経てS281に進み、接続ないし単独と判断されるときはS282に進んでネットワーキング処理を行う。

【0095】図40はその作業を示すサブルーチン・フロー・チャートである。以下説明すると、S400でセグメント終端x座標と自車位置のx座標との差と、検知クラスタx座標と自車位置x座標との差とを求めて比較する。これは、検知クラスタが自車位置に対して対向しているか否か判断するためであり、肯定されるときは自車に接近していると判断してS402に進んでmove(セグメント終端と検知クラスタとの距離を表す)の符号を逆転する。これは、図示しないルーチンで移動距離の変動を見るためである。

【0096】次いでS404に進んでセグメント終端と検知クラスタとの関係が単独であるか否か判断し、肯定されるとき、即ち、検知クラスタがほとんど動いていないとみられるときはS406に進んでセグメント[SNO]マージ回数カウンタ値をインクリメントし、S40

8に進んでカウンタ値が1を超えたか否か判断し、肯定されるときはS410に進んで移動量総和よりセグメント終端位置を減じると共に、移動量2乗和よりセグメント終端位置の2乗値を減じる。これは、図41に示すように、セグメント終端ノードを検知クラスタ側に付け替えるために、それまでのデータを更新する作業を意味する。

【0097】次いでS412に進んで同様にノード数カウンタ値が2を超えたか否か判断し、肯定されたときは曲率(図41に示す)を計算し、S416に進んで曲率総和よりセグメント終端曲率を減じると共に、曲率2乗和よりノード終端曲率2乗値を減じる。これは、S410で説明した図41に示す付け替え作業である。

【0098】次いでS404で否定、即ち、接続または停止とされた場合に備え、S418に進んでセグメントノード数カウンタ値をインクリメントし、S420、S422に進んで同様の処理を行う。次いで図42フロー・チャートのS424に進んで障害物情報ノードメモリを確保する。次いでS426に進んでその確保ができたか否か判断し、肯定されるときはS428、S430、S432を進んで移動体らしさが最低のセグメントを削除してネットワークを圧縮する。また、S426で否定されるときはS434に進んで障害物クラスタ情報をノードに移送する。これは先に図17に示した情報である。

【0099】次いでS436に進み、ノード数が所定保持時間分(例えば2秒分)あるか否か判断し、肯定されるときはS438に進んでセグメント先端の移動量と曲率を各総和より減じると共に、各2乗和も同様に2乗値を減じ、S440に進んでセグメント[SNO]先端へのポインタ値を、セグメント先端の次のノードのポインタ値に書き替える。

【0100】次いでS444に進んで移動体らしさを計算し、S446に進んでセグメントノード長に基づく障害物平均速度計算を行い、S448に進んで障害物移動方位計算を行う。これは、先に述べたネットワークを利用しての最新、最古のノード位置などから速度、方向(方位)を計算することであり、うち速度値、より正確には平均速度値がカルマンフィルタに初期値として与えられる。最後にS450に進んでセグメント[SNO]非接続回数カウンタ値を0にリセットする。

【0101】図38フロー・チャートにおいては、次いでS283に進んで障害物の位置、速度、加速度を推定する。尚、S281で否定されたときはS284に進み、そこで停止と判断されたときはS285に進み、同一であることからセグメント併合処理を行い、S286に進んで同様に障害物位置、速度、加速度推定を行う。S287、S288で検知データクラスタ数、セグメント数がそれぞれ最大値に達したと判断されるまで、上記を繰り返す。

19

【0102】図43ないし44は、障害物位置、速度、加速度推定作業を示すサブルーチン・フロー・チャートである。

【0103】以下説明すると、S500で原点カウンタ $r_{c\ n\ t}$ をインクリメントし、S502に進んでカウンタ値が所定値（例えば20（2000ms分））を超えるか否か判断し、肯定されたときS504に進んでカウンタ値 $r_{c\ n\ t}$ とフラグ $p_{r\ c}$ を0にリセットする。即ち、2秒経過するごとに、障害物移動距離の計測原点を移動（更新）する。

【0104】続いてS506に進んでフラグ $p_{r\ c}$ が0であるか否か判断する。この場合は当然肯定されてS508に進み、フラグ $p_{r\ c}$ に1をセットし、S510に進んでセグメントノード数が1であるか否か判断する。初めてこの処理を行うときは肯定されてS512に進み、図示の如くセグメント終端ノードの障害物位置のx, y座標を用いて原点（Ox, Oy）の移動（更新）を行う。またS510で否定されたときはS514に進み、セグメント終端より1つ前のノードの障害物位置情報を移動距離計測原点にする。

【0105】続いて、S516に進み、前記したカルマンフィルタの値を設定し、S518に進んで前記距離dを移動距離 $x[0]$ の初期値とする。続いてS520に進んで原点カウンタ値 $r_{c\ n\ t}$ が1か否か、即ち、原点更新後最初のカルマンフィルタ計算であるか否か判断し、肯定されるときはS522に進んでセグメント平均速度を速度 $x[1]$ の初期値とする。セグメント平均速度とは、図42フロー・チャートのS446で求めた平均速度を意味する。これは、前記した説明から明らかなる如く、所定期間における平均速度である。

【0106】即ち、実施例に係るカルマンフィルタにおいては、そのモデルに等加速度直線運動モデルを使用し、レーダユニット12を通じて障害物を最初に発見した計測原点からの移動距離を求め、ネットワークから求めた速度を初期値として与えて計測時間ごと（100msごと）にカルマンフィルタの計算を行って100ms後、即ち、次のレーダ検知時刻での障害物の移動距離、速度、加速度を推定している。

【0107】即ち、現在時刻の障害物の位置を推定し、初期値を適切に与えることにより、カルマンフィルタの収束性を向上させ、障害物情報をリアルタイムに得て図3の構成で他のサブシステムに提供することができる。更に、計測原点を2秒ごとに更新して誤差の蓄積を回避している。

【0108】続いてS524に進んでQを零クリアし、S526, S528に進んで共分散行列Q, Rに図示の如き値をセットし、図44フロー・チャートのS530に進んで運動方程式Fの転置行列を求めて値 F_t とし、S532に進んでカルマンフィルタにより、障害物の距離（位置）、速度（初速度）、加速度の推定を行う。こ

こで、Xが推定値である。

【0109】尚、S506でフラグ $p_{r\ c}$ が0ではない、即ち、原点更新が行われないと判断されるときはS534に進んで距離dを図示の如く求める。これは、速度と加速度の初期値として前回推定した値をそのまま用いるので、改めて初期値を設定する必要がないからである。

【0110】次いで図38から図45フロー・チャートに移り、そのS289からS291に進み、セグメント削除判定処理を行う。これは、 $S_{t\ x}[i] = 0$ 、即ち、接続可能な障害物クラスタが1つもなかったセグメント i に関して前記した移動体らしさを所定値と比較し、所定値未満のセグメントを削除する処理である。次いでS292からS294を進んでノード終端位置に基づいてセグメント併合処理を行う。これは、乗用車の平面と同一サイズ内に複数のセグメントが終端を持つとき、それらを併合して1つのセグメントにする処理である。次いでS295を経てS296に進んでネットワーク圧縮処理を行って不要なメモリ領域を開放する。

【0111】図5フロー・チャートに戻ると、次いでS24に進み、障害物情報出力処理を行う。図46はその作業を示すサブルーチン・フロー・チャートである。これは、障害物移動予測テーブルの作成および他システムへの出力作業である。

【0112】以下説明すると、S600において自車位置を入力として道路（走行路）幅Wを算出する。走行路区分線は自車の前後200m分を自車が10m進むごとにリングバッファに読み込み、保持しておくが、それから自車の左側（ないときは右側）の走行路区分線とそれに隣接する区分線を読み出し、道路幅Wを算出する。

【0113】これについて説明すると、追従走行や障害物回避において、障害物がレーダに検知されても、それが道路（走行路）に対してどの位置にあるのか分からなければ、自己の適切な制御は望めない。実施例において、画像処理によって走行路区分線が得られ、画像データ評価部によって不良データの削除、不足データの補填、および道路モデルの識別子が付与される。そのデータを受け、道路のモデルを基準に障害物がどの位置にあるかを統一的に表す。実際は、自己位置の後ろ200mと最新の区分線の地図とモデルのデータを受け、区分線から道路幅を計算する。区分線の最遠の地点よりある程度遠くに障害物があるときは、区分線を2次式で近似する。区分線から障害物までの距離を求め、道路幅と車線位置から障害物の位置を計算する。

【0114】この表現により、追従や回避の判断に必要な自車との位置関係が、単純な演算で可能である。即ち、位置が負のときは1を減算し、その後で少数点以下を切捨て、相手位置-自車位置により自車に対して何車線目にあるかが分かる。図47にそれを示す。更に、走行路区分線に対する障害物の位置を計算する。これによ

り、自車に対する位置を簡単な計算で表すことができ、行動の計画や経路計画に利用することができる。ターゲットの位置を区分線に対する位置として表現する。

【0115】 続いてS602に進んでセグメント数カウンタ値 i をインクリメントし、S604に進み、セグメントノード数NNとノードマージ数NMの和が、しきい値Th（例えば5(500ms)）を超えるか否か、即ち、500ms分のデータが集積したか否か判断する。肯定されときはS606に進み、障害物予測位置、即ち、その移動方位（セグメント傾き）、推定速度、推定加速度に基づいて、所定時刻（例えば100ms）ごとの位置を計算し、障害物移動予測テーブルに書き込む。図48に障害物移動予測テーブルを示す。

【0116】 続いてS608に進んでセグメント i の速度Vが所定値Vth（例えば5km/h）未満か否か、即ち、障害物が移動しているか否か判断し、否定、即ち、移動していると判断されるときはS610に進み、図49に示す如く、障害物移動ベクトルがx軸となるように、走行路区分線GNOデータを回転する。これにより、y座標を参照するだけで対道路（走行路）位置を簡便に求めることができる。

【0117】 続いてS612に進み、障害物位置と予測位置の道路（走行路）に対する位置を求め、障害物移動予測テーブルに書き込む。図50に対道路位置計算を、図51に、障害物位置に対応する区分線データがないときの対道路位置計算を示す。次いでS614に進み、セグメント数カウンタ値 i が最終値に達したことが確認されるまで、上記の処理を繰り返し、確認されるとS616に進み、予測テーブルを外部メモリ（前記した通信用メモリ34）に書き込む。

【0118】 尚、S608で障害物が停止と見做されたときはS618に進んで現在障害物位置に最も近い走行路区分線点列を求め、S620に進んで図示の如く距離dと対走行路位置を求める。

【0119】 図5フロー・チャートに戻ると、S16で検知データがないと判断されるときはS26に進んでネットワークセグメント数カウンタ j をインクリメントし、S28に進んで障害物位置予測を行い（先に図43、44で説明）、S30に進んで障害物情報更新を行う。

【0120】 図52はその作業を示すサブルーチン・フロー・チャートである。以下説明すると、S700でセグメントの非接続回数が所定値（例えば2秒分）を超えたか否か判断し、肯定されるときは、その間見失っていたこととなってデータとして意味をなさないので、S702に進んでセグメントを削除する。これは該当するセグメント情報をクリアしてメモリ領域を空ける作業を意味する。即ち、作業メモリ領域を節約するために、例えばセグメント1、2、3があつた後、セグメント2が消滅し、その後に新たなセグメントが発生したとき、セグ

メント4とせずに、セグメント2とできるようにしておく作業である。

【0121】 S700で否定されるときはS704に進み、先に図40、42で説明したネットワーククリンク処理を行って移動体らしさを計算し、判断する。続いてS706に進み、障害物の位置、速度、加速度の推定（先に図43、44で説明）を行い、S708に進んで障害物情報出力処理（先に図46で説明）を行う。

【0122】 図5フロー・チャートにおいては次いでS10に進んでネットワークセグメント数カウンタ j が最大値に達したことを確認してS34に進み、ネットワーク圧縮処理を行う。これは先にも触れた通り、不要セグメントを削除してメモリ領域を活用する作業である。

【0123】 上記を図6ブロック図を参照して再説すると、移動状態量予測手段（カルマンフィルタ）は、次の検知時刻での障害物の予測位置Posハットnextを求める。また、道路領域境界検出手段（図3の画像評価CPU38）とレーダユニットからの出力に基づき、検知位置データ処理手段は、図10フロー・チャートに20関して述べたクラスタリングを行い、よって得たクラスタごとの障害物位置をPos（絶対座標上の位置、即ち、原点を基準とした位置）として求める。

【0124】 また、障害物時系列情報更新手段は、検出データなどの情報を経時的に更新する。移動状態量予測手段（カルマンフィルタ）は、それに基づいて計測原点を更新すると共に、計測原点からの移動距離x、時系列情報から求めたおよその障害物速度vないしその加速度aに基づいて、次の検知時刻での予測位置Posハットnextを求める。障害物時系列情報記憶手段（ネットワーク型データ構造）では、それら予測値などから障害物の現在位置Pos、予測位置Posハットnext、方位Dir、速度vなどの情報を障害物移動予測テーブル作成手段に送り、そこでテーブル化されて画像処理CPU36などの利用に供される。

【0125】 この実施例は上記の如く構成したので、自車の走行にとって障害となる前走車などの移動体や道路外の静止障害物などを精度良く検出することができる。また前走車などの移動体の情報をリアルタイムに検出し、移動体の運動パラメータを予測するので、衝突の警告ないしは回避動作などをより的確に行うことができると共に、必要に応じて他のサブシステムにその情報を提供することができる。更に、カーブ路（曲線路）などを通行するときも、道路内外の障害物を的確に判別して真に自車の走行の妨げとなるものののみを検出することができる。

【0126】 更に、演算量の少ない直線運動モデルを使用しながら、所定個数のデータを処理したら最新検知位置付近に計測原点を更新するようにしたので、曲線路を含む走行環境においても移動距離の誤差が経時に蓄積することなく、よって予測しようとする位置、速度、

加速度などの運動パラメータの誤差として表れることがない。

【0127】尚、実施例は上記の如く構成したが、画像処理とレーダの検知範囲とは異なるために、障害物の位置によって判定条件を変え、レーダの検知データから求めた移動体らしさに反映させても良い。即ち、検知範囲と検知された位置との関係で、移動体らしさの評価を変えて良い。これにより、移動する障害物と停止している障害物の区別を道路（走行路）の形状を考慮した形で扱うことができる。

【0128】また、レーダと画像処理の検知範囲が異なる場合、特にレーダの検知範囲が画像処理より遠い場合、画像処理の範囲外に検知された障害物は、道路に対する位置が分からることになる。高速道路のような道路形状が長い距離にわたって余り変化しない場合は、過去の（通過してきた）道路構造が変わらない、例えば2車線から3車線への変化がない、ならば、数100m分の自車より後方と前方の最新の走行路区分線データから、最小2乗法により画像処理の範囲外の走行路区分線を補足して障害物の道路（走行路）に対する位置を計算しても良い。

【0129】更に、毎回の画像処理による道路（走行路）の形状も同様なデータが入力されるので、前回結果と道路構造が変わらず、自車の移動量も少ないならば、最小2乗法による画像処理の範囲外の走行路区分線の補足をキャンセルし、前回パラメータを用いて処理を高速化しても良い。

【0130】更に、カルマンフィルタへの初期値として与える速度を単純平均値としたが、加重平均値などでも良い。

【0131】また、適応的に動作する例としてカルマンフィルタを使用したが、それに限られるものではない。

【0132】また、上記において、視覚センサを単眼視としたが、両眼視による距離測定を用いて前方レーダなどを省略しても良い。またレーダユニット12を車両前方のみ設けたが、後方にも設けても良い。また、上記において、「走行路」「道路」「車線」などの名称を混用したが、同一のものを意味することは言うまでもない。

【0133】

【発明の効果】請求項1項においては、自車の走行にとって障害となる前走車などの障害物の位置や移動方向を精度良く検出することができる。それによって、警告、回避など必要な動作を的確に行うことができる。

【0134】請求項2項においては、線形予測フィルタの収束性が向上して推定精度を一層上げることができる。

【0135】請求項3項においては、演算量の少ない直線運動モデルを用いながら、走行環境が曲線路を含むときも、自車の走行にとって障害となる前走車などの移動体の運動パラメータを誤差なく予測することができる。

【図面の簡単な説明】

【図1】この発明に係る移動体の検出装置を備えた車両を全体的に示す説明透視図である。

【図2】図1のレーダの詳細を示す説明図である。

【図3】図1に示すセンサおよびその処理などを詳細に示すブロック図である。

【図4】図3ブロック図の構成をより機能的に示す図2と同様の説明図である。

【図5】この発明に係る移動体の検出装置の動作を示す10メインフロー・チャートである。

【図6】図5フロー・チャートと同様にこの発明に係る移動体の検出装置の動作を説明する機能ブロック図である。

【図7】図5フロー・チャートの（障害物の）クラスタ化を説明する説明図である。

【図8】図1の車両がカーブ路を通過するときのレーダビームと障害物との関係を説明する説明図である。

【図9】実施例に係る走行路区分線によるクラスタの切りわけを説明する説明図である。

【図10】図5フロー・チャートの検知データのクラスタ化作業を示すサブルーチン・フロー・チャートである。

【図11】図10フロー・チャートで予定する走行環境の説明図である。

【図12】図10フロー・チャートの作業を説明する説明図である。

【図13】同様に図10フロー・チャートの作業を説明する説明図である。

【図14】同様に図10フロー・チャートの作業を説明する説明図である。

【図15】同様に図10フロー・チャートの作業を説明する説明図である。

【図16】この実施例で使用する移動情報のネットワークセグメントを示す説明図である。

【図17】図16でのノードの情報を示す説明図である。

【図18】図16でのネットワークセグメントの情報を示す説明図である。

【図19】この実施例で用いた推定速度と推定方位を示すシミュレーションデータ図である。

【図20】図16のネットワークセグメントの単純更新を説明する説明図である。

【図21】同様に図16のネットワークセグメントの複数障害物が存在するときの更新を説明する説明図である。

【図22】この実施例で用いる移動体らしさなる概念を表現する要素の説明図である。

【図23】その移動体らしさの算出手法を示す説明図である。

【図24】その移動体らしさの推移、前走車、停止障害

25

物を示すシミュレーションデータ図である。

【図25】図5フロー・チャートのネットワーク更新処理作業を示すサブルーチン・フロー・チャートである。

【図26】図25フロー・チャートの障害物位置予測作業を示すサブルーチン・フロー・チャート部分である。

【図27】この実施例での移動予測による位置推定を説明する説明図である。

【図28】図27の移動予測手法による、実際の移動する車両における推定位置と入力位置とを示すシミュレーションデータ図である。

【図29】同様に図27の移動予測手法による、停止障害物における推定位置と入力位置とを示すシミュレーションデータ図である。

【図30】前走車が定速から減速して停止する際の実際の前走車の速度、この実施例における推定速度、1秒間の予測位置における位置を示すシミュレーションデータ図である。

【図31】この実施例におけるカルマンフィルタの移動体モデルを示す説明図である。

【図32】この実施例におけるカルマンフィルタの初期値として、真値 50 km/h に対して初期値 100 km/h を与えた場合の収束状況を示すシミュレーションデータ図である。

【図33】同様に、この実施例におけるカルマンフィルタの初期値として、セグメントの最新、最古のノードから求めた値を与えた場合の収束状況を示すシミュレーションデータ図である。

【図34】図1に示す車両がカーブ路を走行するときの計測距離を説明する説明図である。

【図35】カルマンフィルタの計算での原点更新をした場合としない場合の推定速度の優劣を示すシミュレーションデータ図である。

【図36】この実施例での「停止」「単独」「接続」「非接続」の概念を説明する説明図

【図37】図25フロー・チャートに続くフロー・チャート部分である。

【図38】図37フロー・チャートに続くフロー・チャート部分である。

【図39】図38フロー・チャートのネットワーク発生処理を示す説明図である。

【図40】図38フロー・チャートのネットワーキング

26

ク処理作業を示すサブルーチン・フロー・チャートの前半部である。

【図41】図40の作業を説明する説明図である。

【図42】図40フロー・チャートの後半部である。

【図43】図38フロー・チャートの障害物位置、速度、加速度推定作業を示すサブルーチン・フロー・チャートの前半部である。

【図44】図43フロー・チャートの後半部である。

【図45】図38フロー・チャートに続く最終フロー・チャート部分である。

【図46】図5フロー・チャートの障害物情報出力処理作業を示すサブルーチン・フロー・チャートである。

【図47】図46フロー・チャートの道路に対する障害物の位置表現を示す説明図である。

【図48】図46フロー・チャートの障害物移動予測テーブルを示す説明図である。

【図49】図46フロー・チャートの走行路区分線データの回転作業を説明する説明図である。

【図50】図46フロー・チャートの対道路位置計算を示す説明図である。

【図51】図46フロー・チャートの障害物移動予測テーブルへの書き込み処理作業を示す説明図である。

【図52】図5フロー・チャートの障害物情報更新作業を示すサブルーチン・フロー・チャートである。

【符号の説明】

10 CCDカメラ (モノクロTVカメラ)

12 レーダユニット

14 ヨーレートセンサ

16 車速センサ

18 舵角センサ

30 画像処理ハードウェア

36 画像処理CPU

38 画像評価CPU

40 レーダ処理回路

42 レーダ評価CPU

44 軌跡推定CPU

46 軌跡追従制御CPU

50 行動計画意思決定CPU

52 舵角制御CPU

40 60 車速制御 (加速度制御) CPU

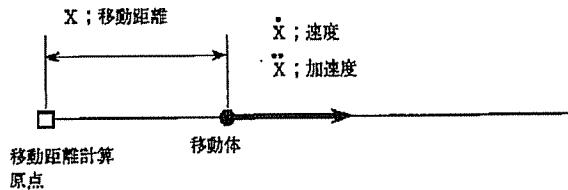
【図23】

$$\text{移動体らしさ} = w_1 \cdot A_1 + w_2 \cdot A_2 + w_3 \cdot B + w_4 \cdot C$$

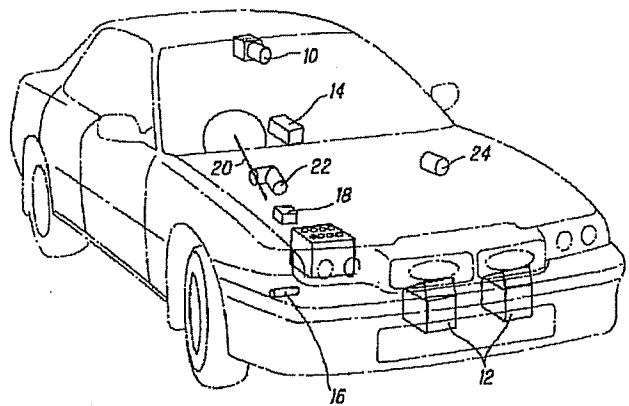
$$\text{但し } w_1 + w_2 + w_3 + w_4 = 1$$

$$\begin{aligned} \text{例: } & w_1 = w_2 = w_3 = 0.3 \\ & w_4 = 0.1 \end{aligned}$$

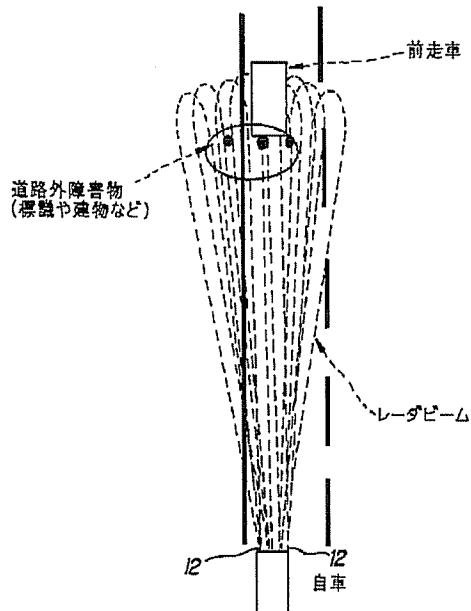
【図31】



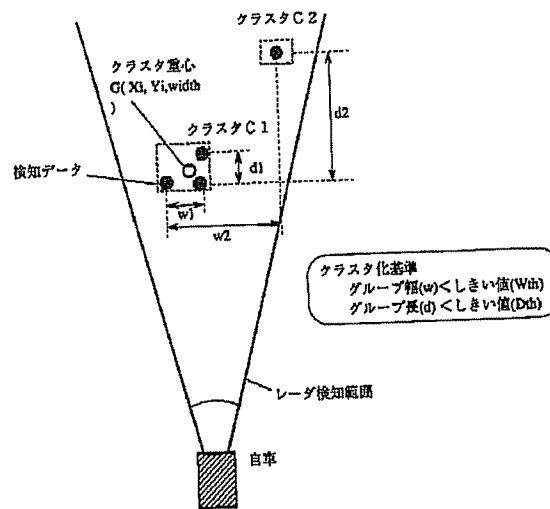
【図1】



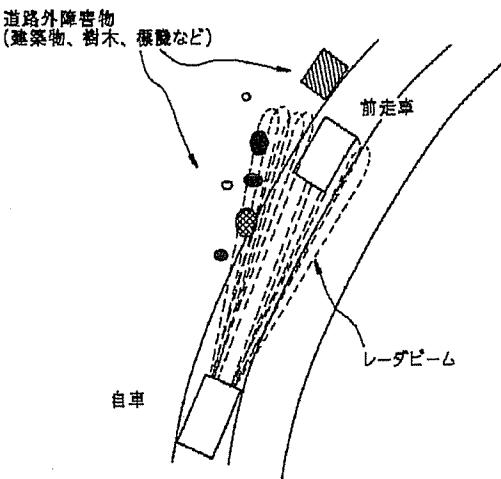
【図2】



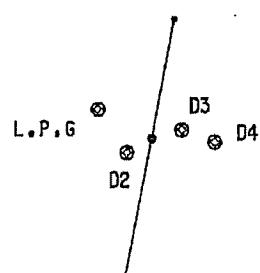
【図7】



【図8】



【図12】



【図17】

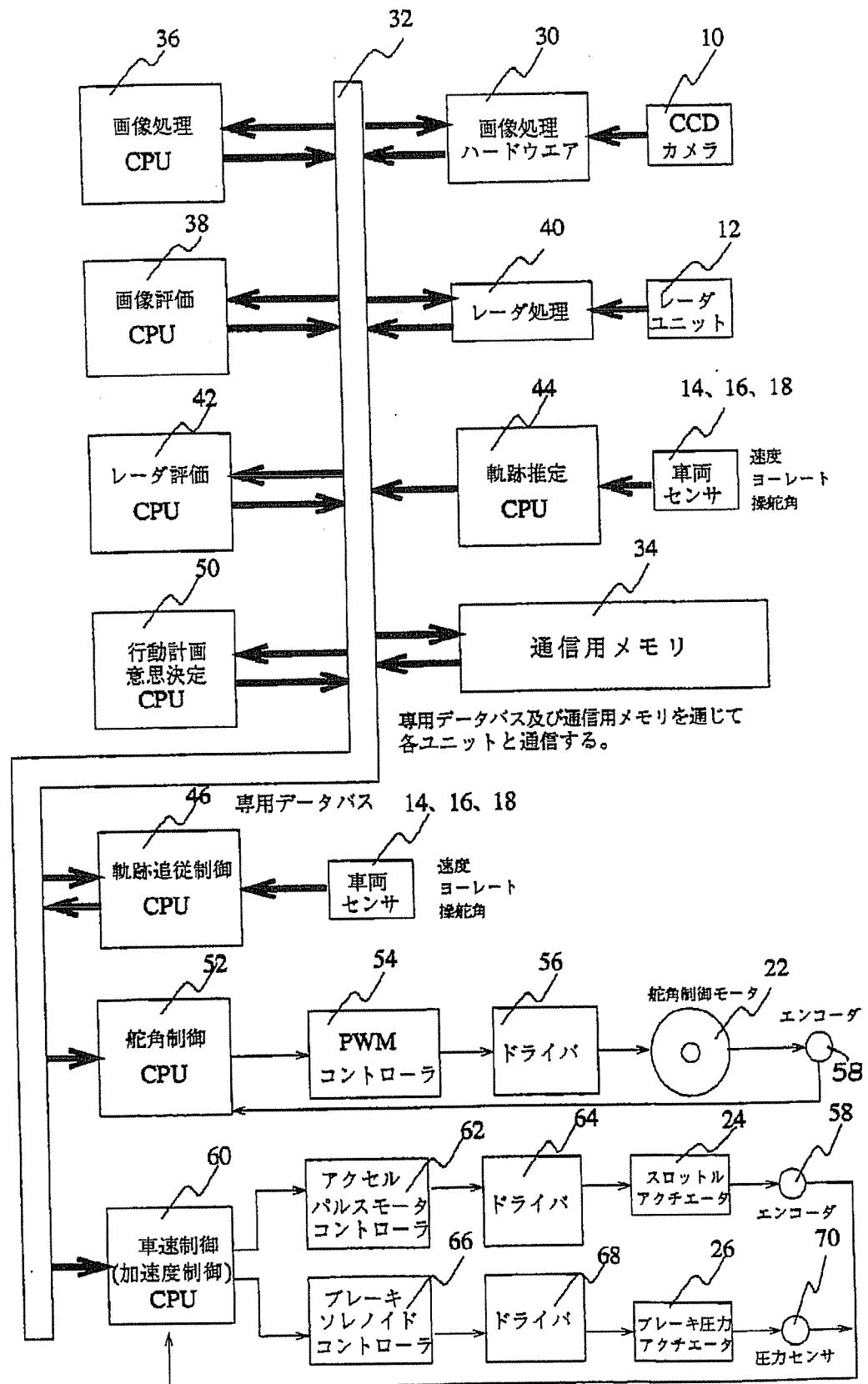
移動体属性へのポインタ	下図の「移動体属性情報」
検知時刻	レーダ検知時刻
移動曲率	1つ前と2つ前のノードおよび本ノードと1つ前のノードのなす角
移動量	1つ前と本ノードとの距離
速度	現在推定速度
方位	現在の推定進行方向
走行路に対する位置	自像処理結果の区分線走行路モデルに対する位置
次のノードへのポインタ	

ノードの情報

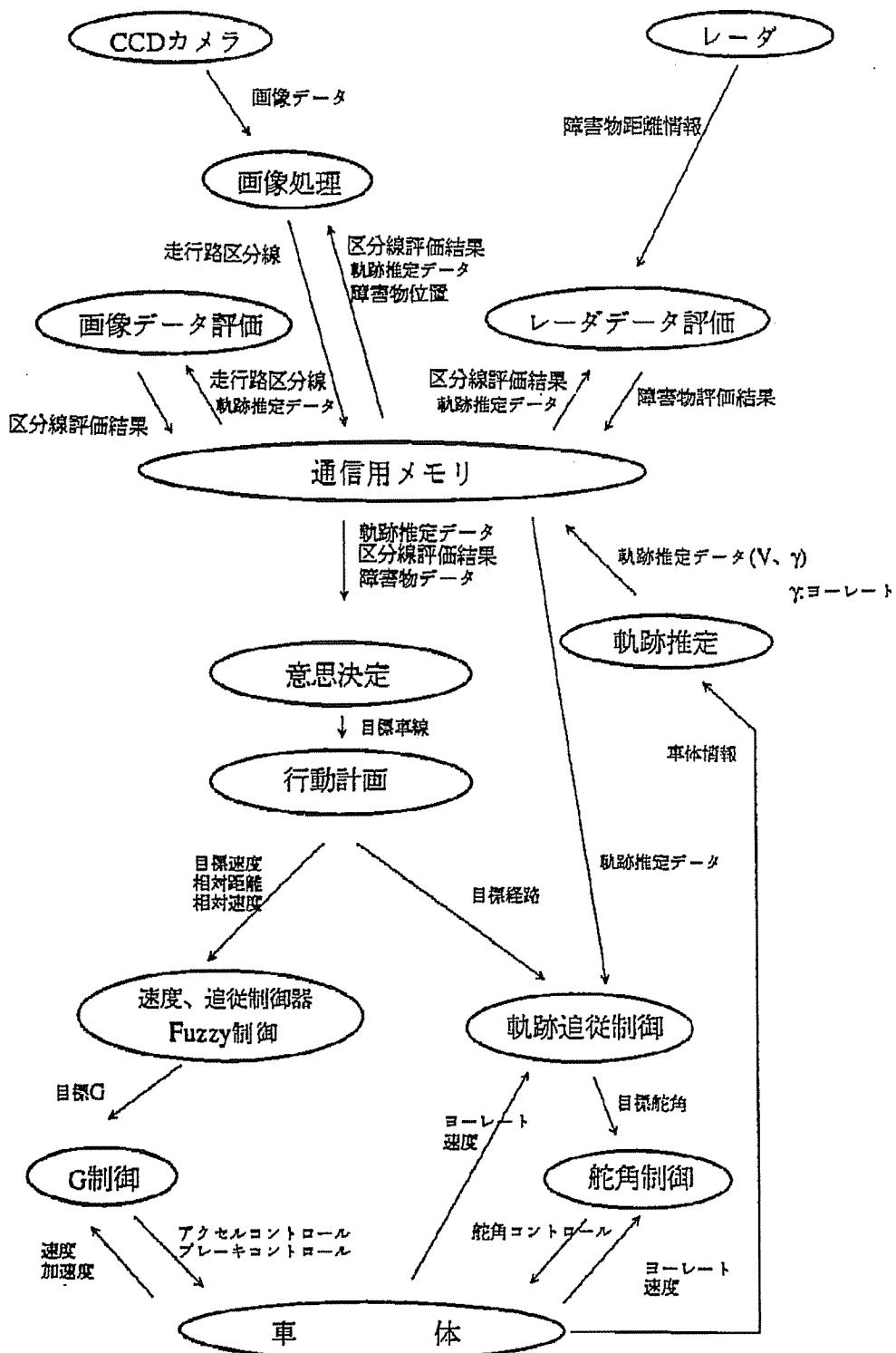
位置座標	現在位置(クラスタ重心)絶対座標
幅	クラスタ幅
高さ	高さ方向がわかる レーダを想定
相対位置座標	レーダ検知位置

移動体属性情報

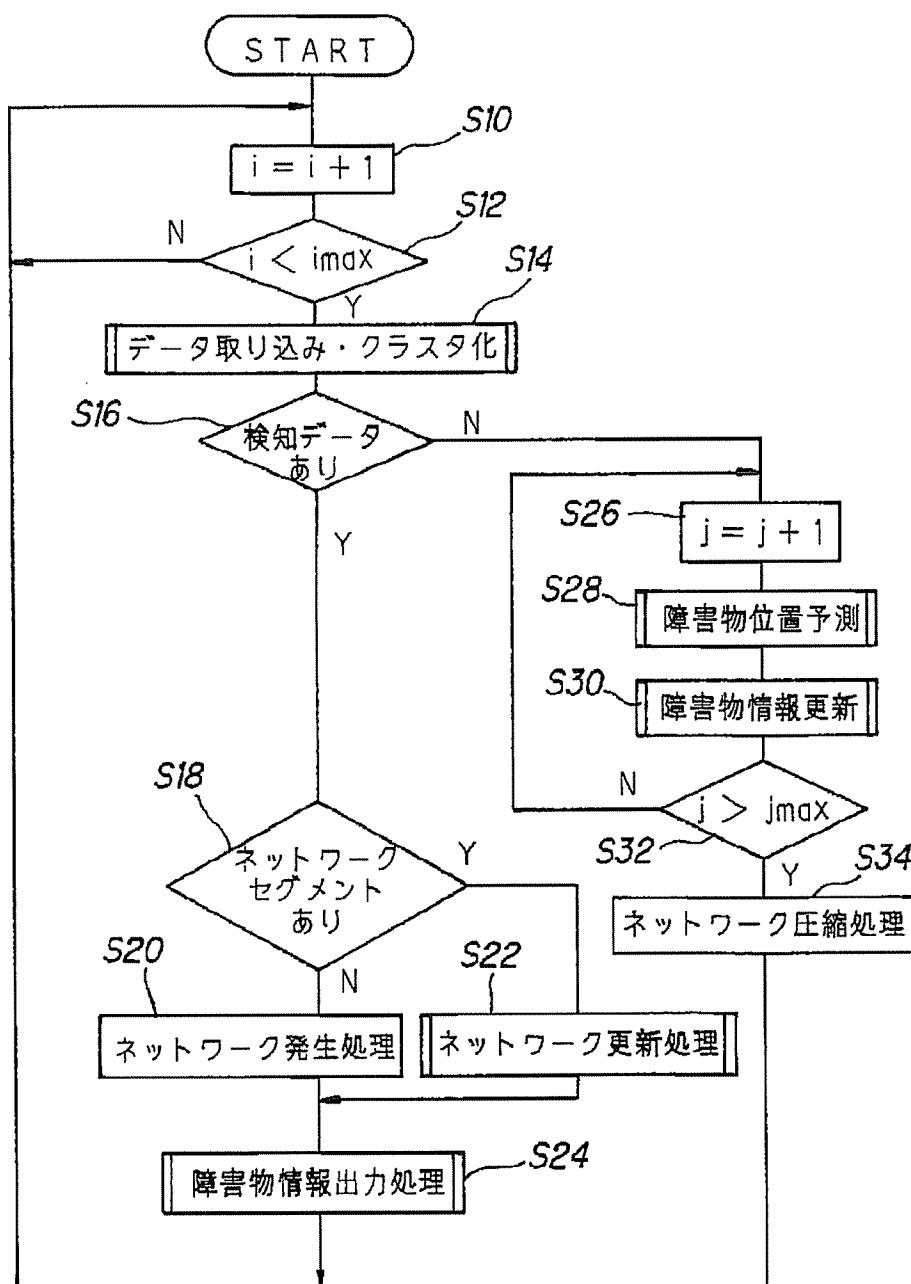
【図3】



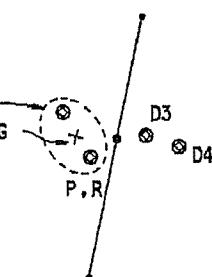
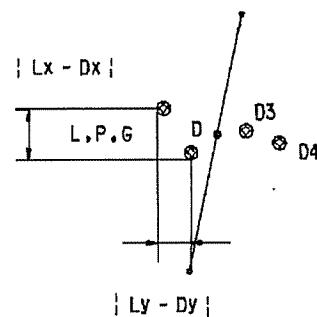
【図4】



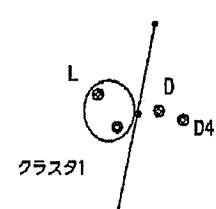
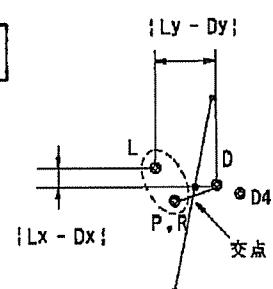
【図5】



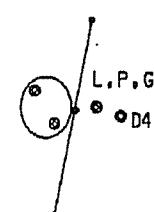
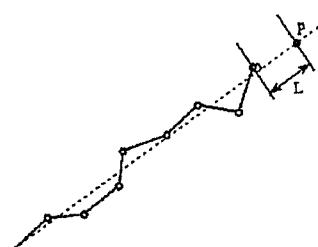
【図13】



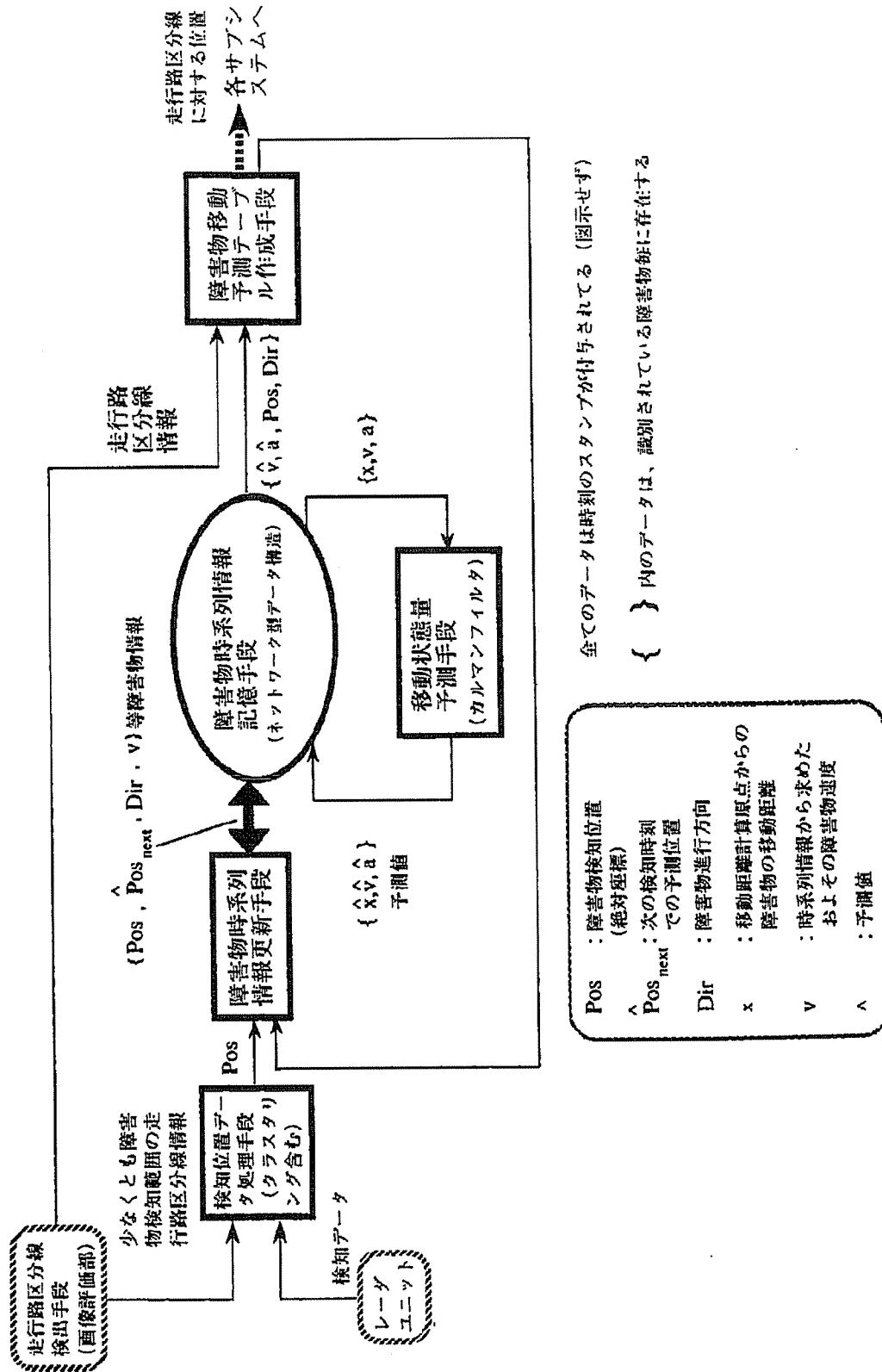
【図14】



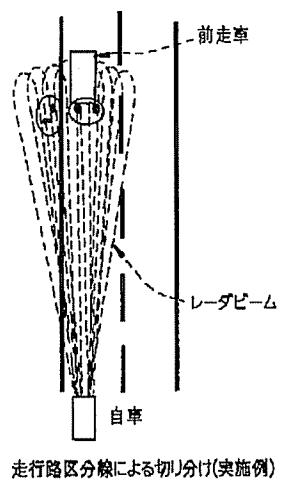
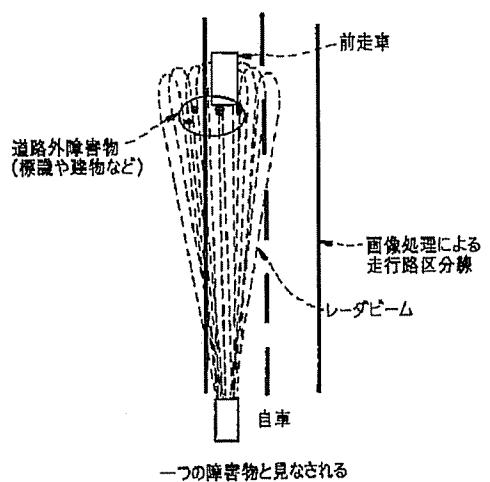
【図27】



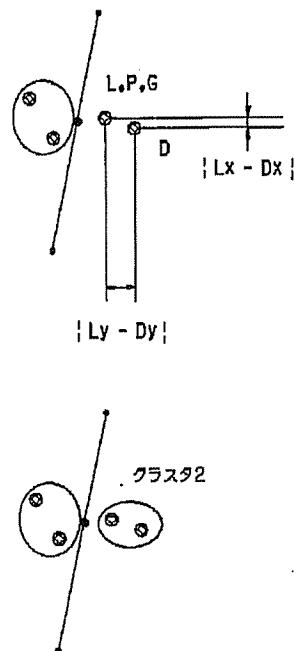
【図6】



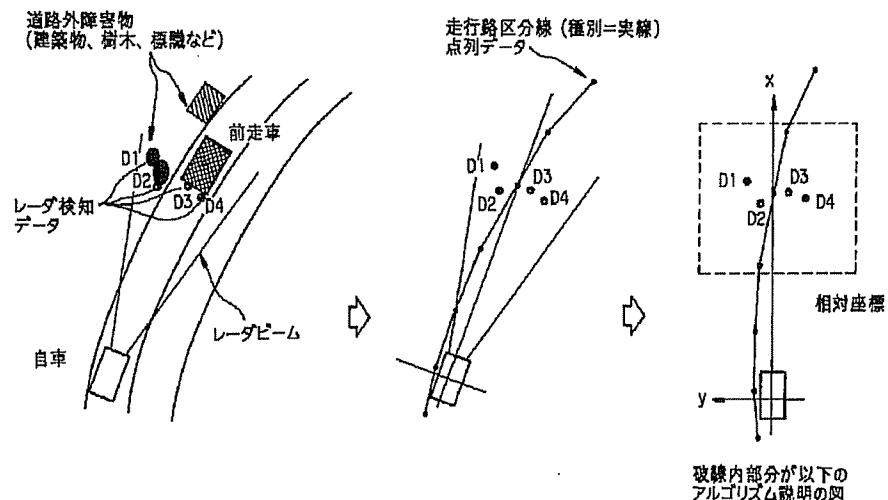
【図9】



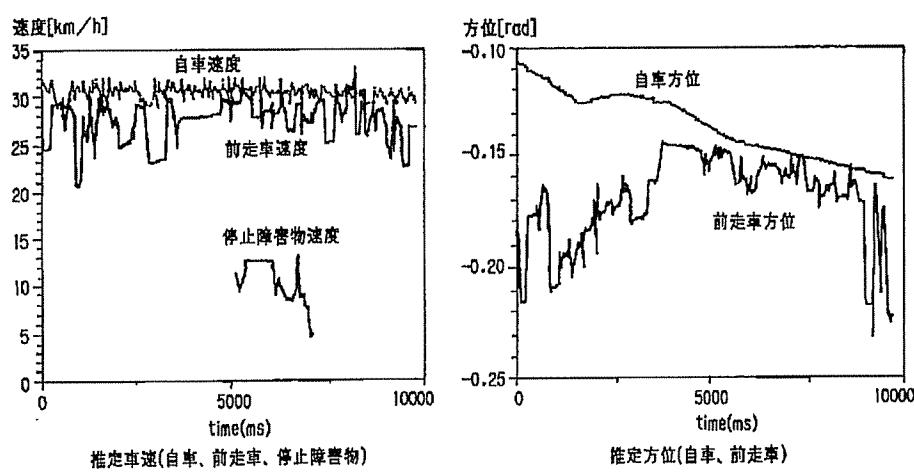
【図15】



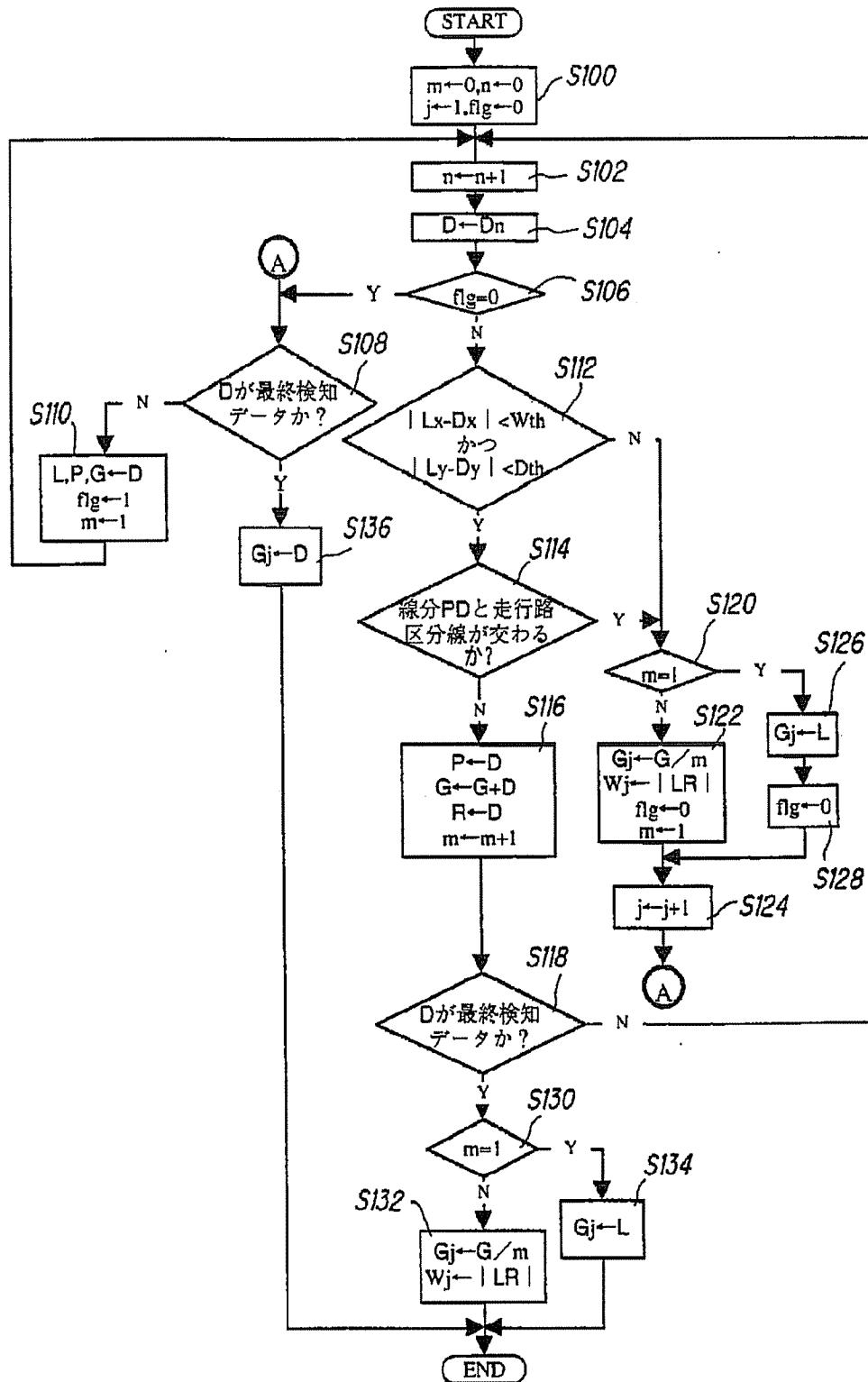
【図11】



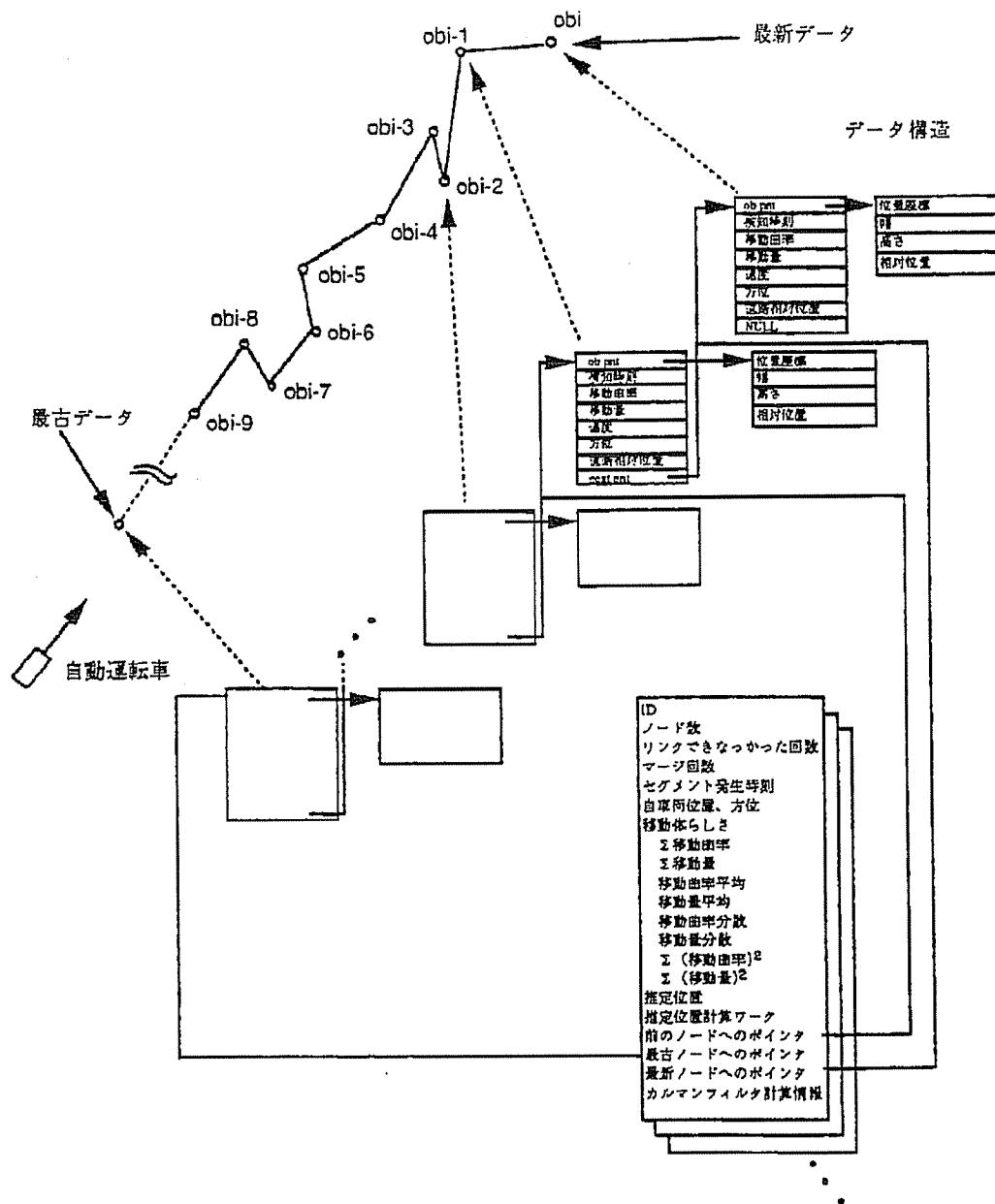
【図19】



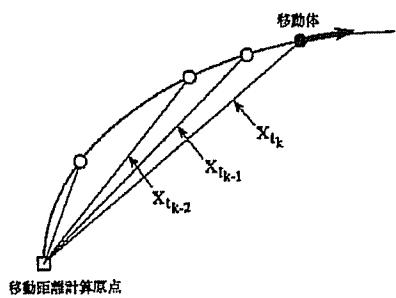
【図10】



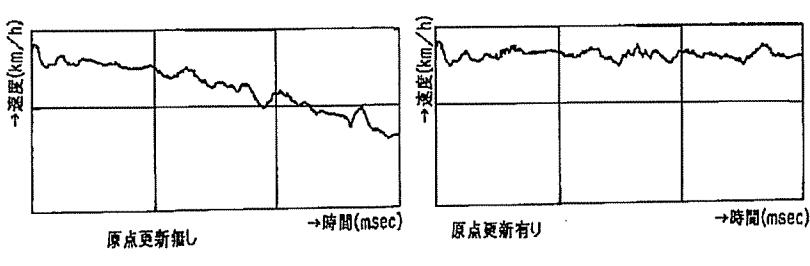
【図16】



【図34】



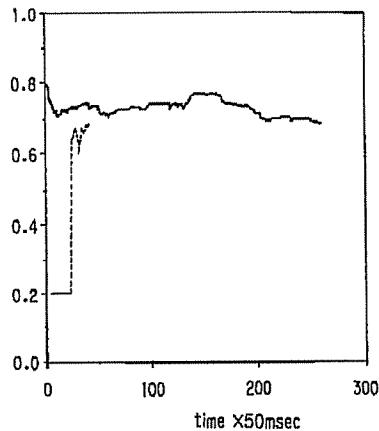
【図35】



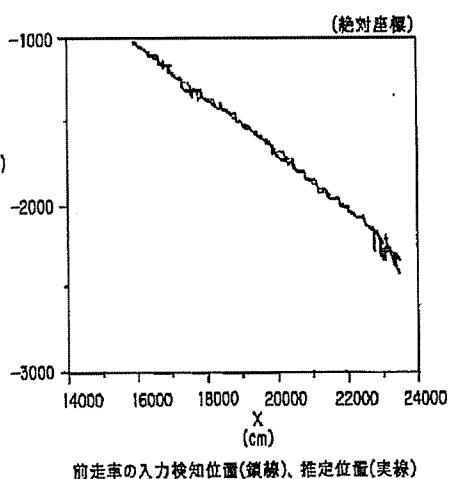
【図18】

セグメントID	障害物固有識別子
ノード数	ネットワーク構成ノードの総数
リンクリンクできなかった回数	間違ノードが無かった（検知状態の連續性が無くなかった）場合に計数
マージ回数	停止していると見做された回数
セグメント発生時刻	初めて検知された時刻
自車位置、方位	軌跡推定に基づく現在の自車位置
移動体らしさ	後述
セグメント移動曲率総和	ノード曲率の総和
セグメント移動量総和	ノード移動量の総和
セグメント移動曲率平均	ノード曲率総和／ノード数
セグメント移動量平均	ノード移動量総和／ノード数
セグメント移動曲率分散	曲率二乗和／ノード数 - (曲率平均) ²
セグメント移動量分散	移動量二乗和／ノード数 - (移動量平均) ²
セグメント移動曲率二乗和	$\Sigma (\text{曲率})^2$
セグメント移動量二乗和	$\Sigma (\text{移動量})^2$
推定位置	最新ノード推定位置
前のノードへのポインタ	
最古ノードへのポインタ	
最新ノードへのポインタ	
カルマンフィルタ計算値	計算原点移動処理用カウンタ、同計算用フラグ、推定値（距離、速度、加速度）、カルマンゲイン、誤差分散、距離計測原点、距離

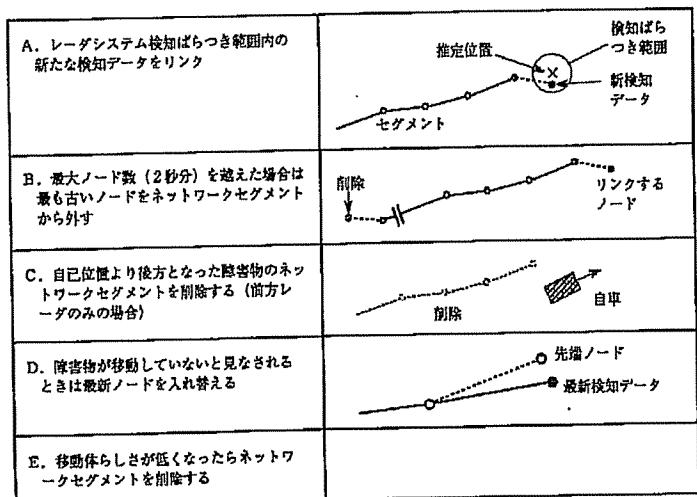
【図24】



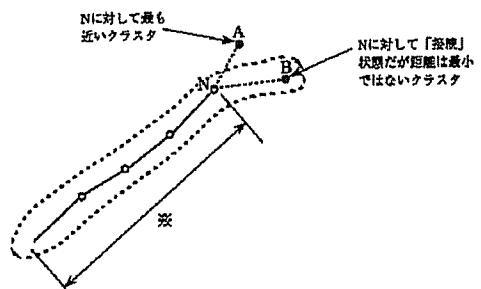
【図28】



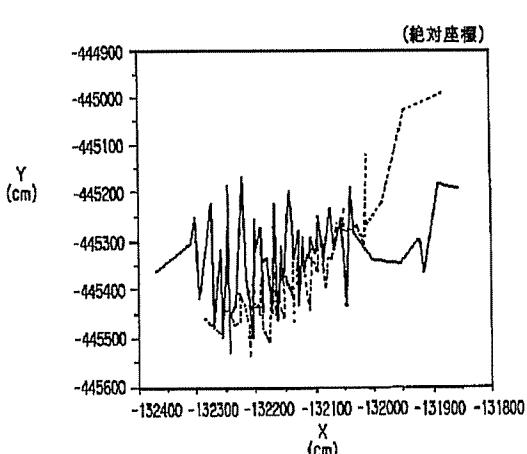
【図20】



【図39】



セグメントの分岐継承
セグメントの過去の部分(※印)をコピーしたものに、Bのクラスタを追加して新しいセグメントとする

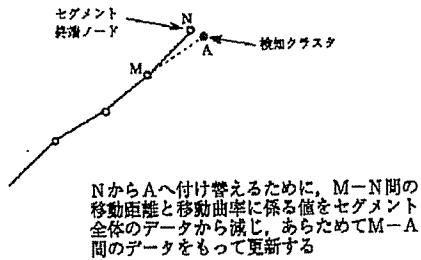


停止障害物の検知位置(虚線)、推定位置(実線)

【図21】

更新条件	処理内容	用
A. 新規発生 他の障害物が初めて現れた (CASE1) L2>レーダ検知ばらつき範囲かつ L1<L2	セグメント1 セグメント2 (新規セグメントとして登録)	■→ 自車 □→ 自分より転送を車両を新たに選択した
B. 分岐检测 他の障害物が初めて現れた (CASE2) L1<L2<レーダ検知ばらつき範囲	セグメント1 セグメント2 (セグメント1の情報を維持する)	■→ □→ どちらかが車線変更した
C. 合流消滅 複数通った障害物が合流した 複数のセグメントの終端同士が所定距離内に接近したとき	セグメント1 セグメント2 移動体らしさの小さい方のセグメントを削除	■→ □→ 障接車線からの割り込みがあった

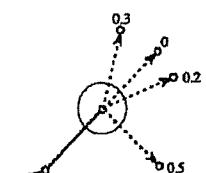
【図41】



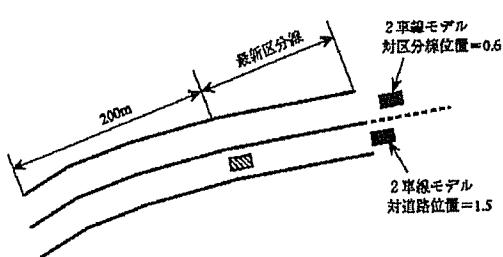
【図22】

評価項目	評価値
移動の滑らかさ	$A_1 = \text{移動曲率平均}$ $A_2 = 1/\sqrt{\text{移動量分散}}$
長い時間レーダに捕捉されている、或いは検知不能時間が短い	$B = \frac{(\text{現在ノード数}) - (\text{検知されなかったサイクル数})}{(\text{現在ノード数})}$
センサ (CCD, レーダ) の検知範囲と障害物との位置関係	P : レーダの検知範囲 Q : 画像処理によって走行路区分線を検出可能な範囲 と定めるとき、障害物の位置が P ∩ Q に有る → C = 1.0 P - (P ∩ Q) に有る → C = 0.5 Q - (P ∩ Q) に有る → C = 0 (実施例では検知不可) 走行路外 に有る → C = 0 

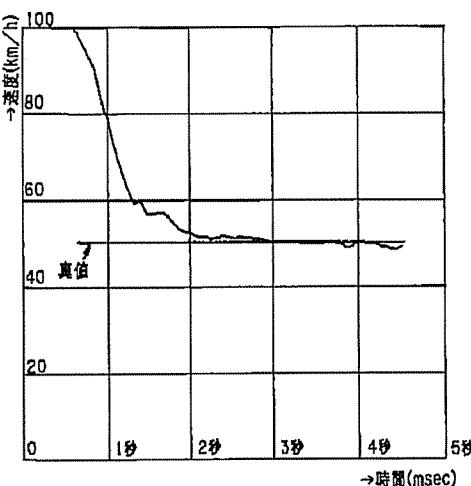
但し、移動曲率は隣接するノード間での移動方向変化分であり、180度を零、0度を1と定める。



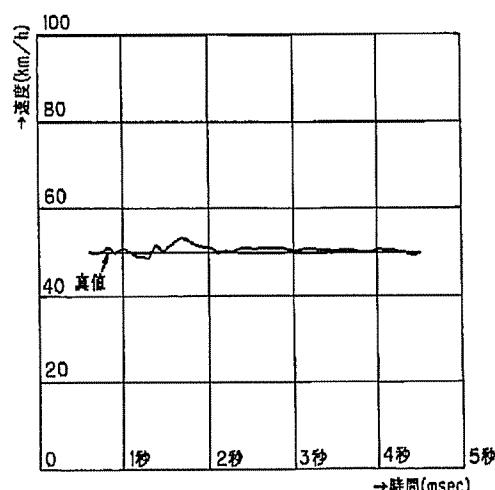
【図47】



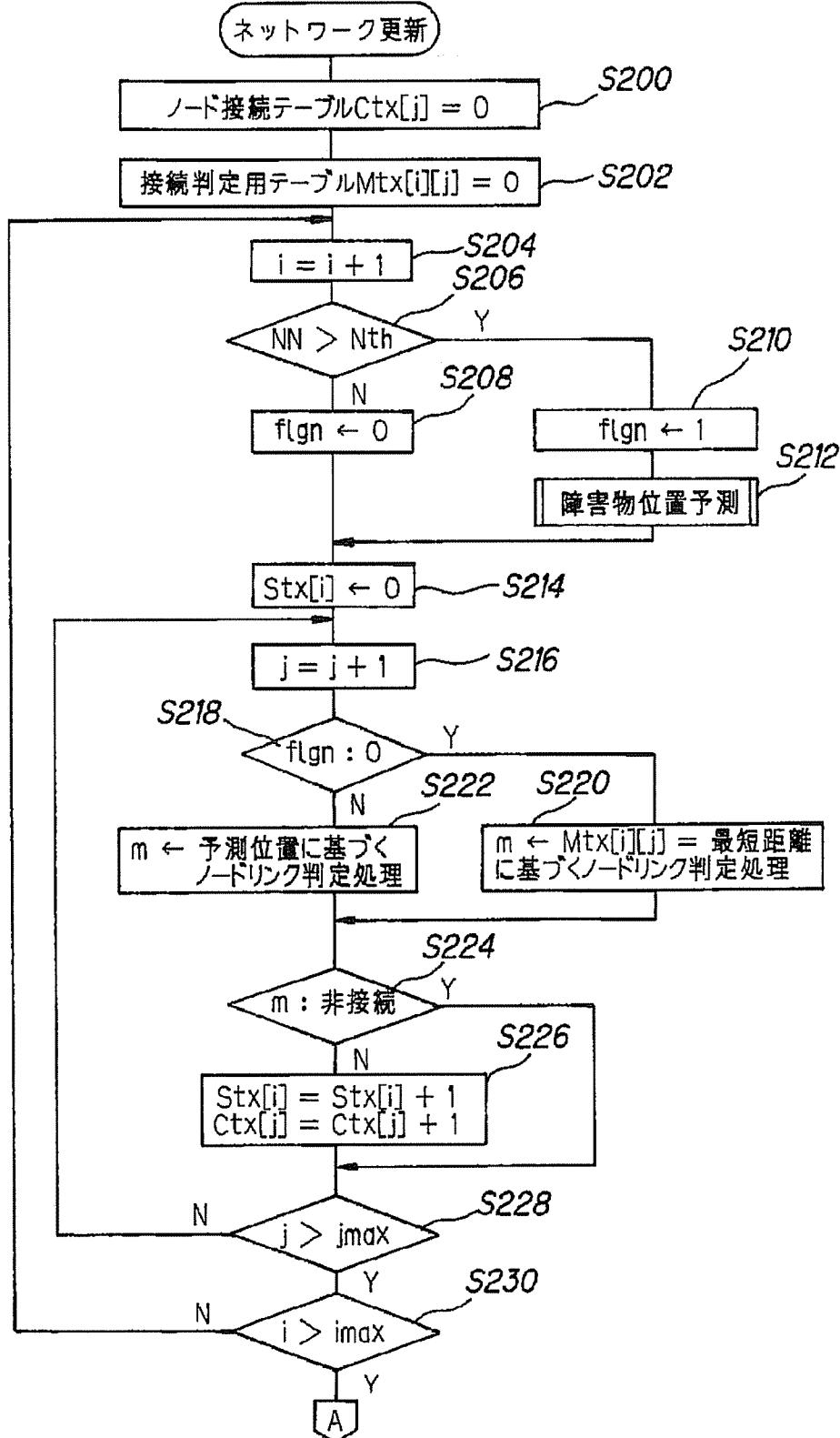
【図32】



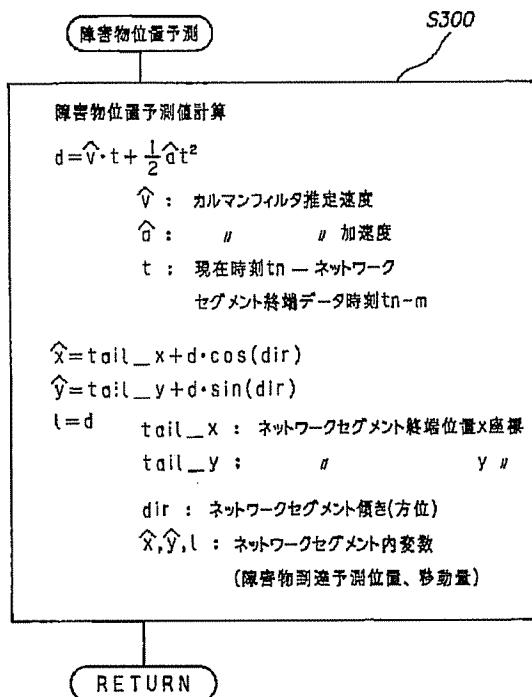
【図33】



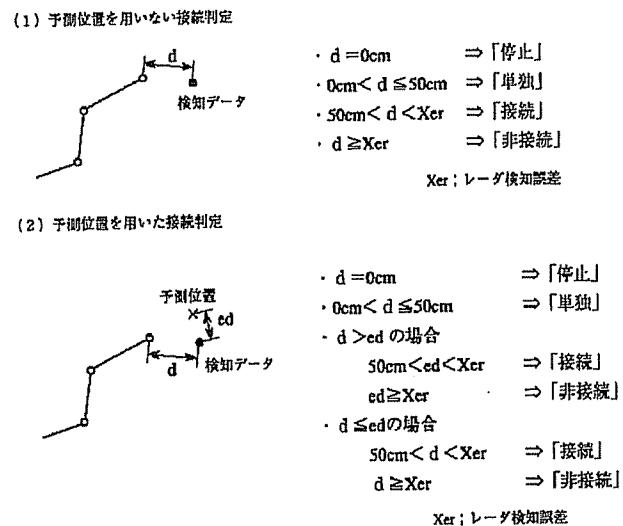
【図25】



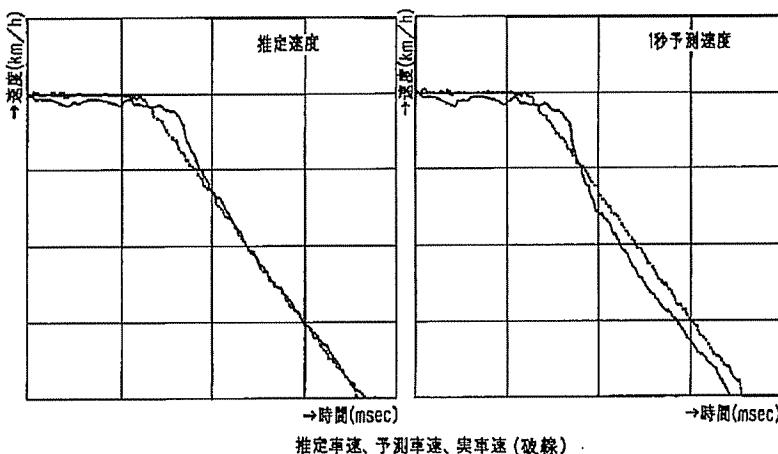
【図26】



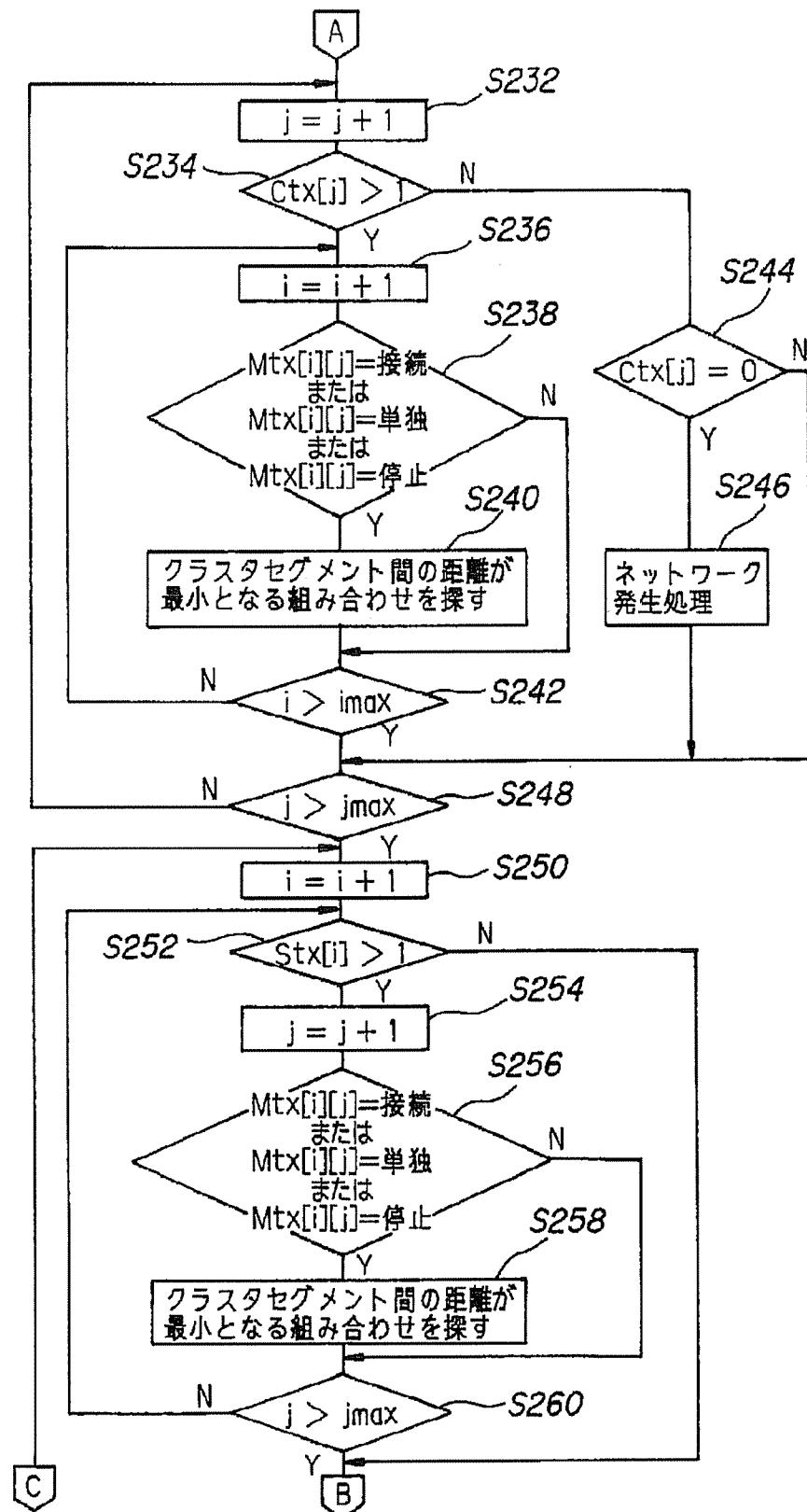
【図36】



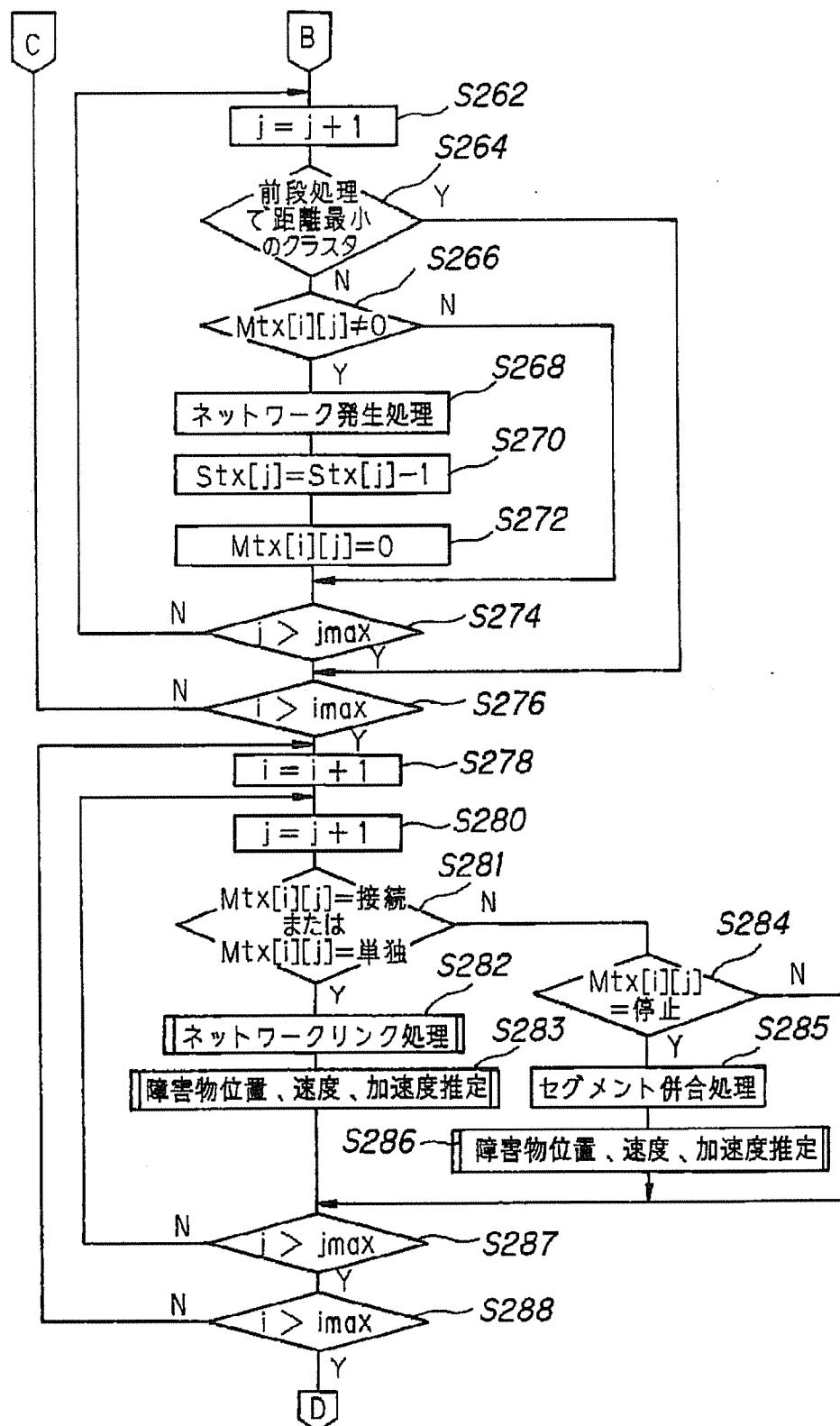
【図30】



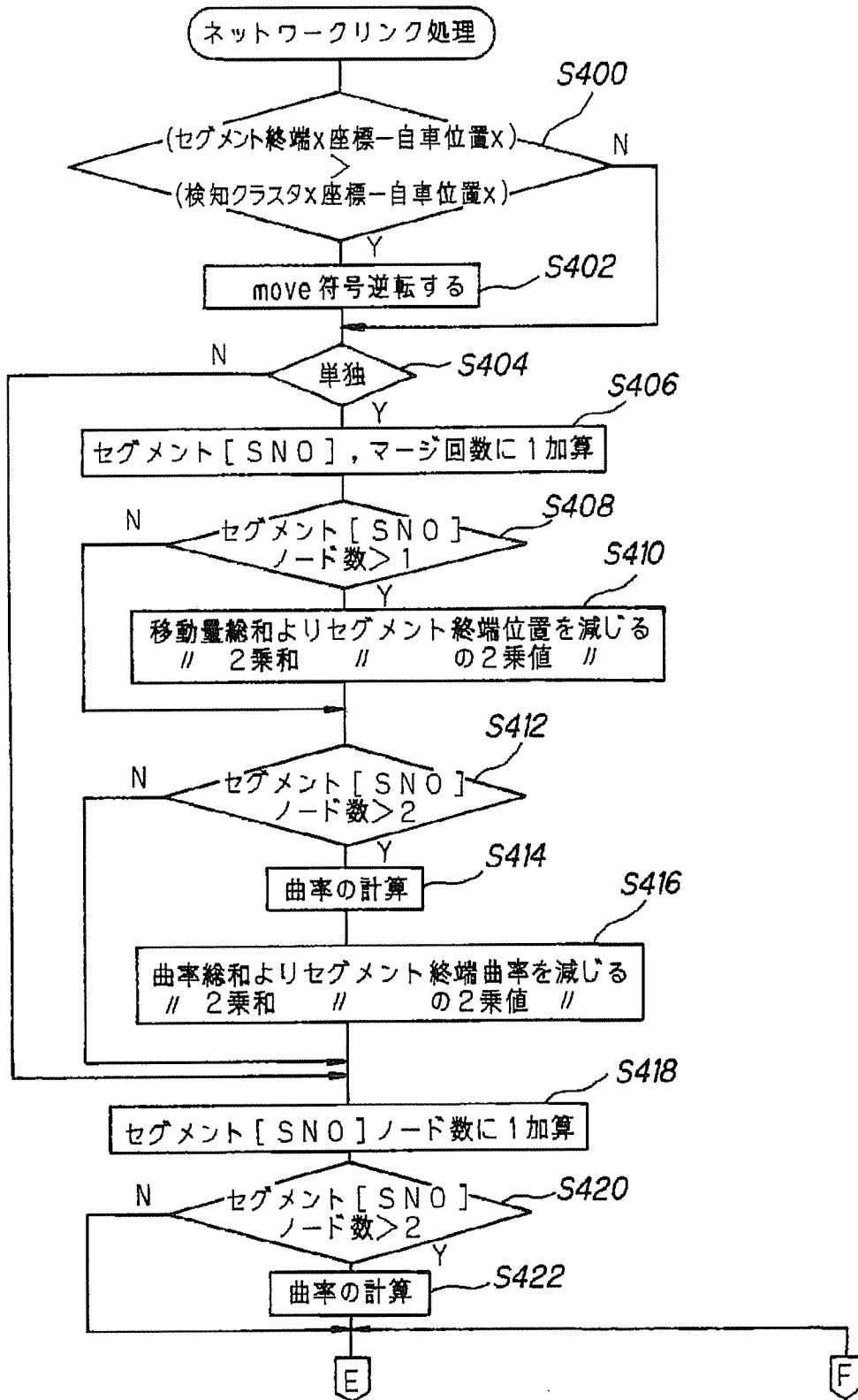
【図37】



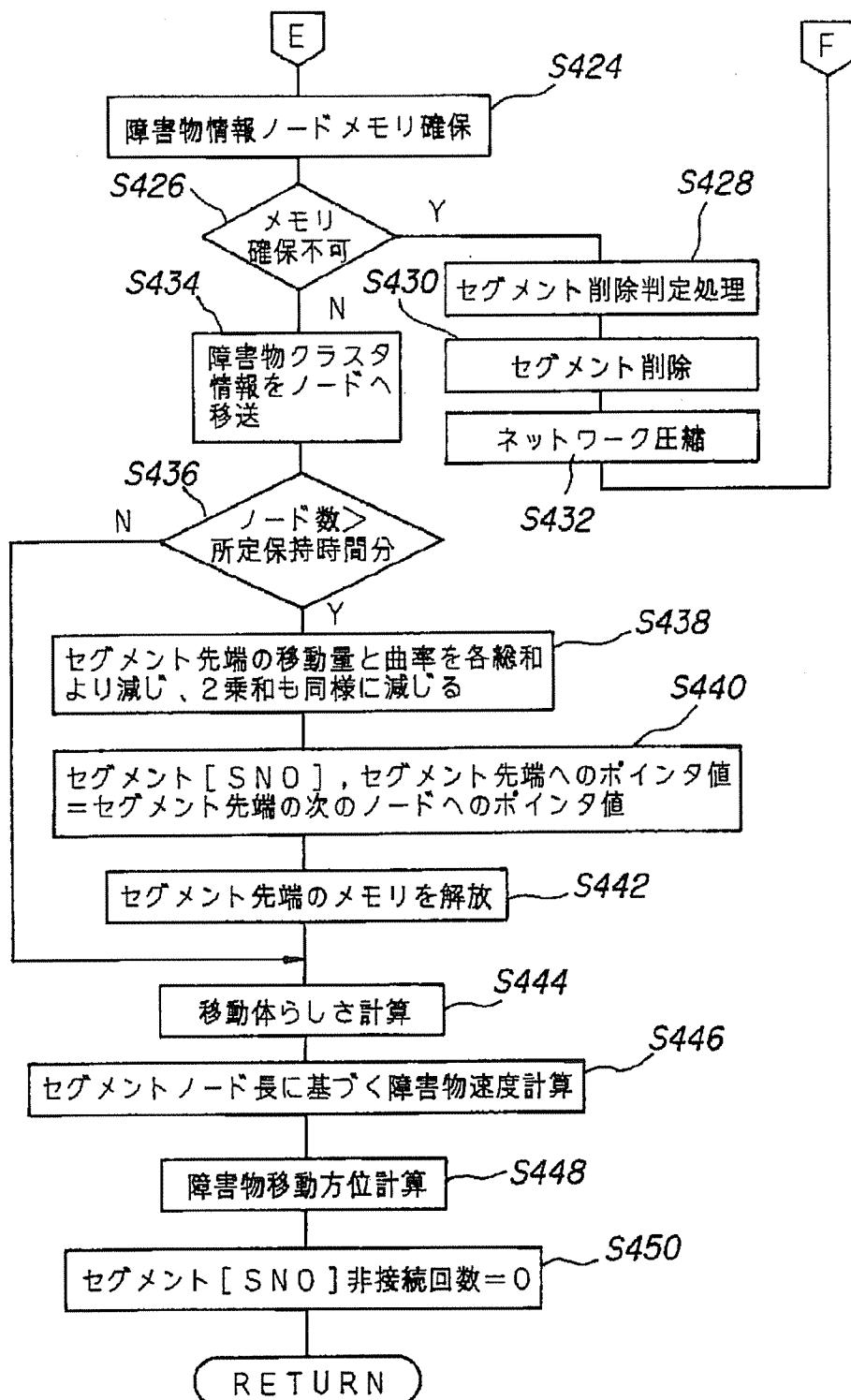
【図38】



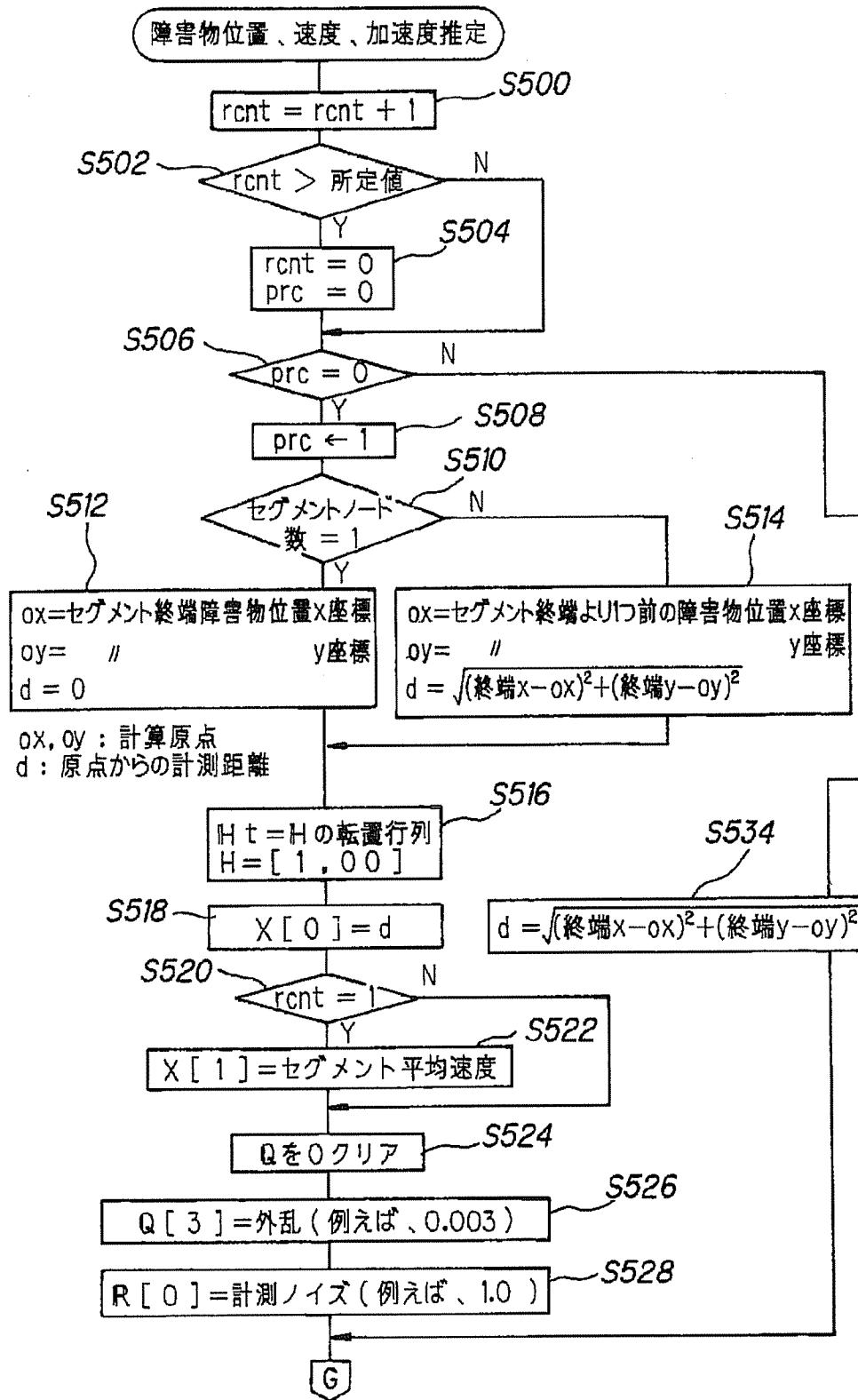
【図40】



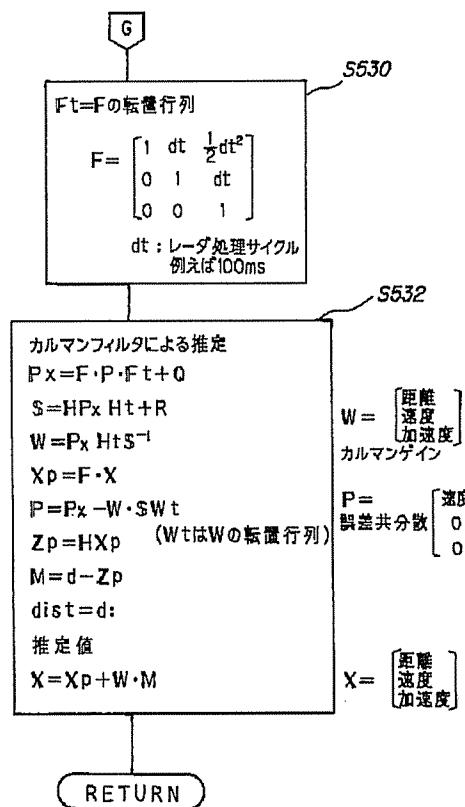
【図42】



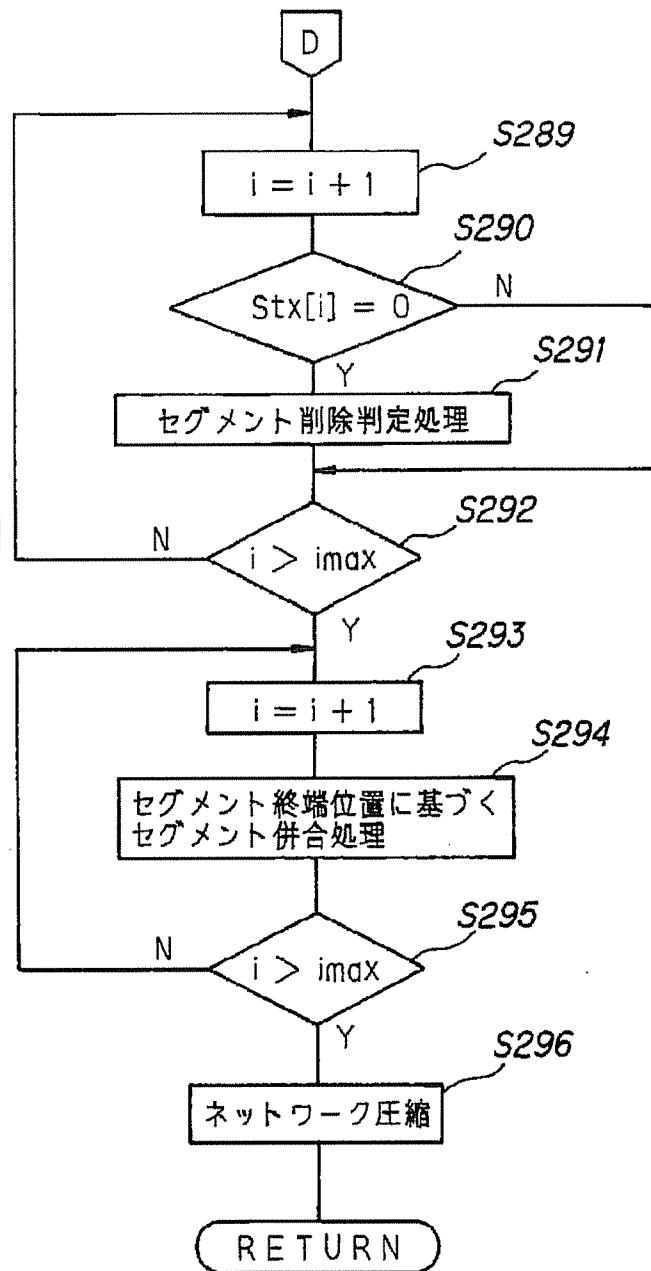
【図43】



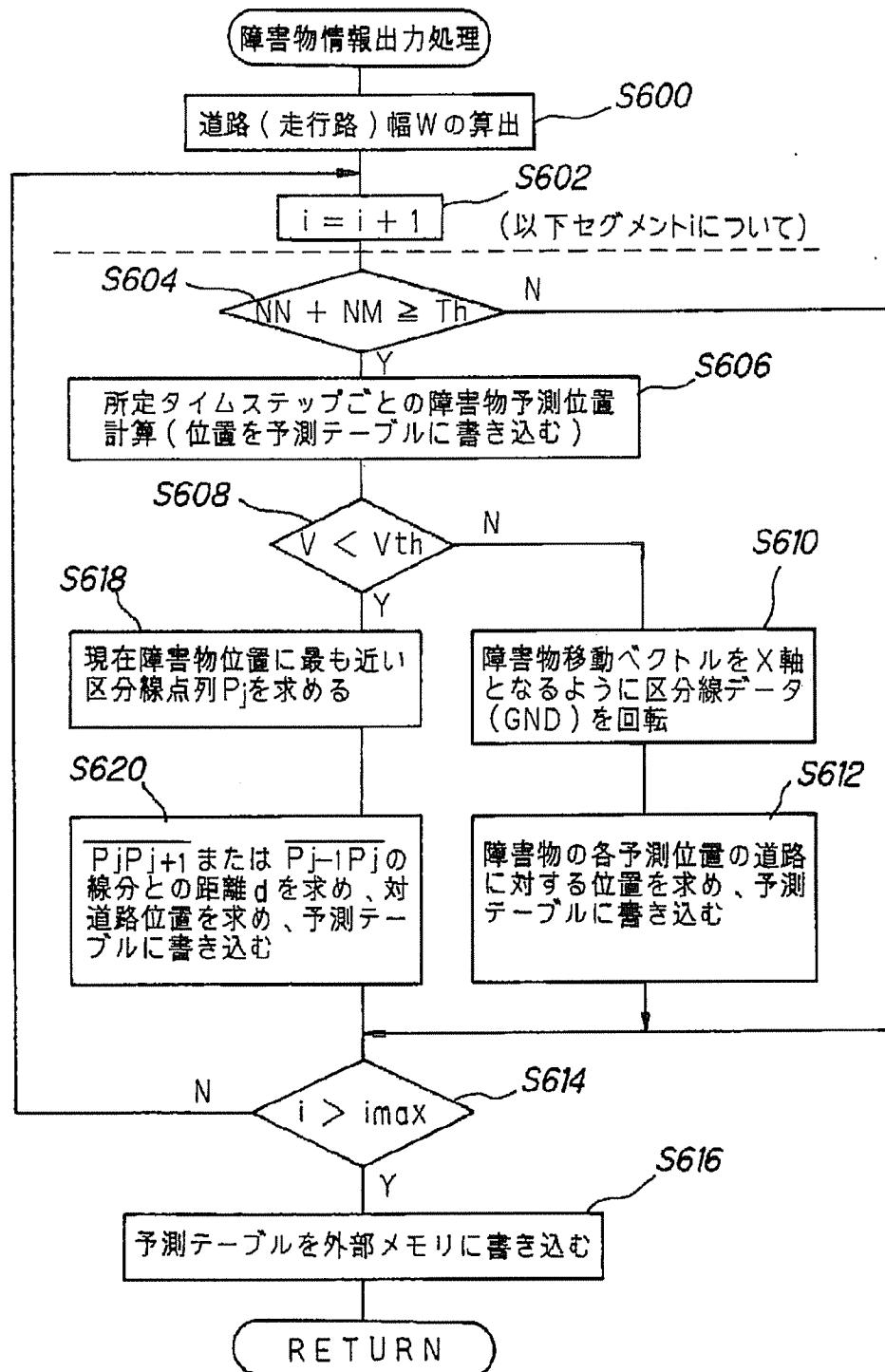
【図4-4】



【図4-5】



【図46】

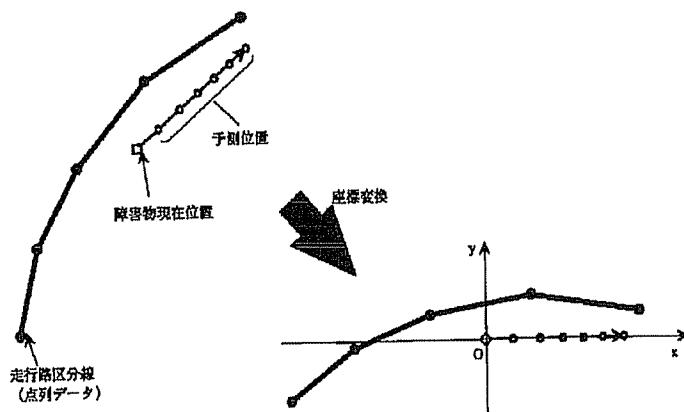


【図48】

障害物ID	x座標	y座標	速度	走行路区分線に対する位置	
現在					
100ms後					
200ms後					
300ms後					
400ms後					
500ms後					
600ms後					
700ms後					
800ms後					
900ms後					
1000ms後					
1000ms後					
900ms後					
1000ms後					

現在提えている障害物の数だけ作成する

【図49】

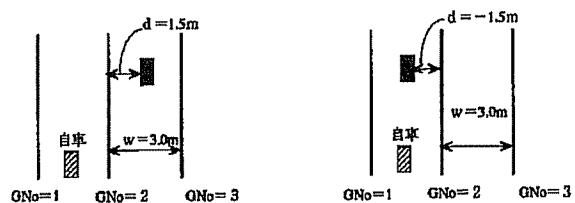


【図50】

対区分線位置(Pos)計算式

$$Pos = \frac{\text{走行路区分線までの距離} d}{\text{走行路幅} w} + \text{区分線番号} GNo$$

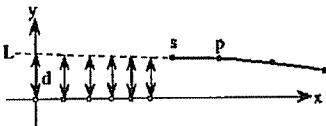
(自車のすぐ左の区分線をGNo=1とする)



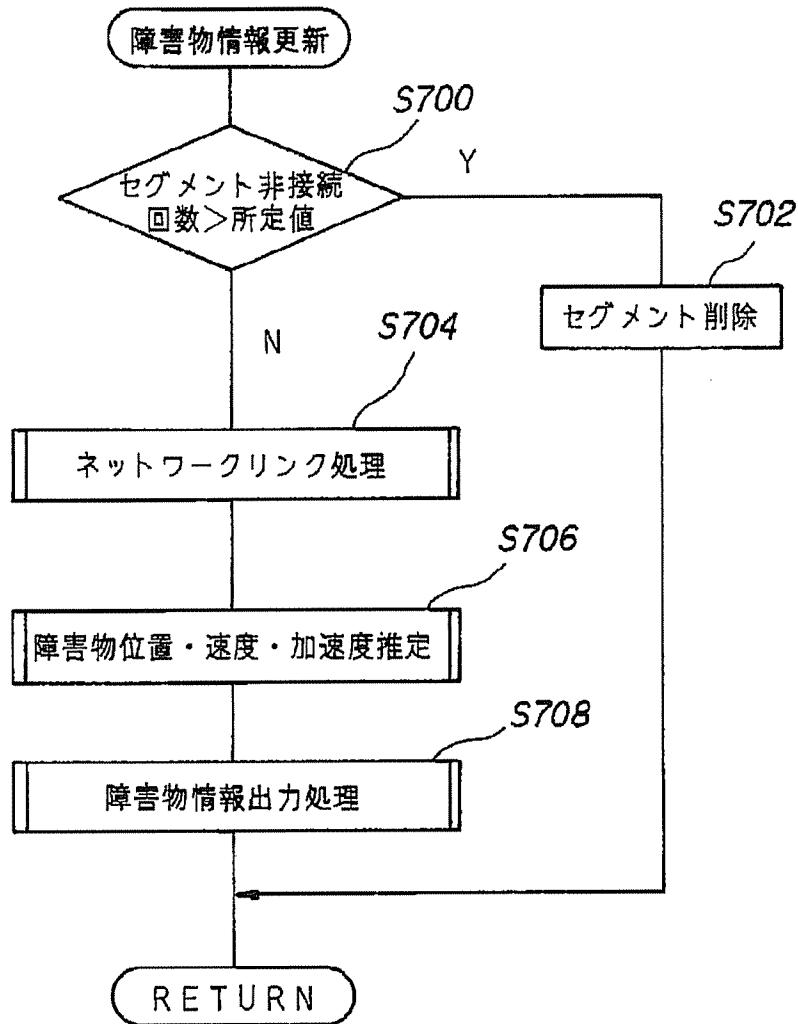
$$Pos = 1.5 / 3.0 + 2 = 2.5$$

$$Pos = -1.5 / 3.0 + 2 = 1.5$$

【図51】

①区分線が障害物より後方にある		区分線始端 s とその次の点 p を結ぶ線分 L に対する、各障害物位置からの距離に基づき対区分線位置を求める
②区分線が障害物より前方にある		

【図52】



フロントページの続き

(51) Int.Cl. ⁶	識別記号	序内整理番号	F I	技術表示箇所
G 0 8 G	1/16	A		
H 0 4 N	7/18	J		